UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Vitomir Banjac

Preplet standardov varovanja informacij pri procesiranju plačilnega prometa

MAGISTRSKO DELO

Mentor: izr. prof. dr. Marjan Krisper

Ljubljana, 2016

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Večna pot 113
1000 Ljubljana, Slovenija
telefon: 01 47 98 100
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Številka: 136-MAG-ISO/2016
Datum: 29. 02. 2016

Vitomir BANJAC, univ. dipl. inž. rač. in inf.

Ljubljana

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Preplet standardov varovanja informacij pri procesiranju plačilnega prometa**

Information security standards in payment card industry

Tematika naloge:

Ena od ključnih dejavnosti združb, ki procesirajo občutljive podatke, je (tudi) varovanje informacij. Da bi se združbe lažje spopadale z varovanjem informacij, obstaja kopica standardov, ogrodij in dobrih praks, po katerih se lahko ali morajo ravnati. Ti standardi so si v svojih zahtevah lahko različni, nekatere zahteve in poglavja pa so si lahko podobna. V magistrskem delu bo opisal teoretično ozadje informacijske varnosti. Predstavil bo implementacijo standarda PCI DSS in ISO/IEC 20000 z uporabo ITIL za primer podjetja v dejavnosti procesiranja plačilnega prometa. Implementacijo bo predstavil tudi na primerih tveganj in ranljivosti, modeliranih v jeziku ArchiMate. Kot glavni cilj magistrskega dela želi preveriti, kako bi lahko podjetje s kar najmanj stroški in porabe različnih virov udejanjilo še standard ISO/IEC 27001 na podlagi tega, kar je na voljo iz že prej omenjenih standardov. Naredil bo pregled in primerjavo ter preslikavo zahtev med omenjenimi standardi. Predlagal bo tudi idejni koncept modela implementacije ISO/IEC 27001 z integracijo PCI DSS in ITIL, s katerim lahko podjetje zmanjšanja stroške in preprosteje implementira vpeljavo novega standarda ter zmanjšanja nivo tveganj.

Mentor:

izr. prof. dr. Marjan Krisper

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

I Z J A V A O A V T O R S T V U

magistrskega dela

Spodaj podpisani/-a Vitomir Banjac,

z vpisno številko 63990026,

sem avtor/-ica magistrskega dela z naslovom

Preplet standardov varovanja informacij pri procesiranju plačilnega prometa

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal/-a samostojno pod vodstvom mentorja (naziv, ime in priimek)

  izr. prof. dr. Marjan Krisper

- so elektronska oblika magistrskega dela, naslova (slov., angl.), povzetka (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela

- in soglašam z javno objavo elektronske oblike magistrskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 26.5.2016 Podpis avtorja/-ice: Banjac

Zahvaljujem se izr. prof. dr. Marjanu Krisperju za pomoč pri izdelavi magistrske naloge.

Zahvaljujem se podjetju Bankart.

Hvala staršem in bratu za dobre nasvete.

Posebna hvala moji družini, Sabini in otrokoma Tilnu in Tiborju, za potrpežljivost in lepe trenutke.

Tatiju

Kazalo

Seznam uporabljenih kratic in simbolov

ATM - Automated Teller Machine

CAB - Change Advisory Board

CDS - Card Data Security

CMDB - Configuration Management Database

COBIT - Control Objectives for Information and Related Technology

CUIT - Coded User Interface Testing

DAM - Database Activity Monitoring

ENISA - European Network and Information Security Agency

ISO/IEC - International Organization for Standardization/International Electrotechnical Commission

ISMS - Information Security Management System

ITIL - Information Technology Infrastructure Library

ITSM - Information Technology Service Management

OWASP - Open Web Application Security Project

PCI DSS - Payment Card Industry Data Security Standard

PAN - Primary Account Number

PDCA - Plan-Do-Check-Act

POS - Point of Sale

SEPA - Single Euro Payments Area

SIMP - SEPA infrastruktura za mala plačila

SQL - Structured Query Language

SSL - Secure Sockets Layer

SUVI - Sistem za upravljanje varovanja informacij

TLS - Transport Layer Security

WSDL - Web Services Description Language

Seznam tabel

Seznam slik

Povzetek

Naslov: Preplet varovanja informacij pri procesiranju plačilnega prometa

Varovanje informacij je pomembna dejavnost mnogim združbam, še posebej če deluje v okolju z občutljivimi podatki. Za lažjo implementacijo varovanja informacij je na voljo mnogo standardov, ogrodij in dobrih praks, po katerih se združbe lahko ali pa se morajo ravnati. Ti standardi so si v svojih zahtevah lahko različni, nekatere zahteve in poglavja pa so si lahko podobna.

Proučeno je teoretično ozadje informacijske varnosti v procesiranju plačilnega prometa, izbrani in proučeni so standardi in ogrodja, ki nudijo pomoč pri varovanju informacij. Predstavljena je implementacija standarda PCI DSS in ISO/IEC 20000 z uporabo ITIL za primer podjetja v dejavnosti procesiranja plačilnega prometa. Implementacija je predstavljena tudi na primerih tveganj in ranljivosti, modeliranih v jeziku ArchiMate. Glavni cilj magistrskega dela je preveriti, kako bi lahko podjetje s kar najmanj stroški in porabe različnih virov udejanjilo še standard ISO/IEC 27001 na podlagi tega, kar je na voljo iz zahtev standardov PCI DSS in ITIL. Narejen je pregled, primerjava in preslikava zahtev med omenjenimi standardi. Podana je tudi zamisel modela implementacije ISO/IEC 27001 z integracijo PCI DSS in ITIL, s katerim lahko podjetje zmanjšanja stroške in preprosteje udejanji vpeljavo novega standarda ter zmanjša nivo tveganj. V zaključku so oblikovani končni sklepi in ugotovitve ter predlogi za nadaljnje delo.

V magistrskem delu je uporabljeno znanje, pridobljeno pri podiplomskem magistrskem študiju Informacijski sistemi in odločanje na Fakulteti za računalništvo in informatiko ter znanje in izkušnje, pridobljene pri delu na področju razvoja programske opreme, predvsem spletnih aplikacij in spletnih storitev, ter implementaciji varnostnih standardov v združbo, ki je obravnavano v magistrskem delu. Znanje se poleg omenjenega črpa predvsem iz tujih in domačih znanstvenih, strokovnih člankov, prispevkov na konferencah, standardov in ogrodij.

Ključne besede: informacijska varnost, standard, PCI DSS, ITIL, ISO/IEC 27001, integracija

Abstract

Title: Information security standards in payment card industry

Data security is an important activity in many companies, especially if they operate in an environment with sensitive data. To facilitate the implementation of data security measures, a variety of standards, frameworks and best practices are available as a guidelines according to which a company can or must act. These standards can be different in their requirements, while some of their requirements and chapters can be similar.

This work examines theoretical background of information security in the processing of payments, while selected standards and frameworks that help to safeguard information are also analysed. The case of implementation of the standard PCI DSS and ISO/IEC 20000 using ITIL in companies in the business of processing payment transactions is presented. Additionally, implementation in cases of risk and vulnerability, modelled in the ArchiMate language, is also demonstrated. The main aim of the master thesis is to examine how the company could at minimum cost and use of various sources implement standard ISO/IEC 27001 on the basis of what is already available from the standards PCI DSS and ITIL. For this purpose, the master thesis reviews, compares and conducts mapping of various requirements between those standards. On this basis, the concept of the implementation model of ISO / IEC 27001 with the integration of the PCI DSS and ITIL is developed. Through this model, companies could lower their costs and more easily implement the new standard as well as reduce the level of risk. The conclusion of the thesis offers overview of the findings and suggestions for further work.

In this master thesis knowledge acquired in postgraduate study of Information Systems and decision-making at the Faculty of Engineering and Computer Science is used. Moreover, this thesis makes use of the knowledge and experience gained from my work in the field of software development, in particular web applications and web services, as well as the implementation of safety standards in the network which is discussed in this thesis. In addition to this, domestic and foreign scientific, technical articles, conference contributions, standards and frameworks are used as a relevant knowledge sources.

Keywords: information security, standard, PCI DSS, ITIL, ISO/IEC 27001, integration

1 Uvod

Podatki in informacije so pri poslovanju marsikaterega podjetja ključnega pomena, še posebej če gre za občutljive in zaupne podatke strank. Varovanje podatkov je tako lahko ena od ključnih (podpornih) dejavnosti takega podjetja, saj z morebitno zlorabo lahko pride do ogromne poslovne škode, tako v smislu ugleda kot tudi v finančnem smislu. Varovanje (lahko) predpisujejo varnostni standardi in ogrodja, katerih certifikate mora podjetje imeti (tudi) zaradi zahtev strank. Podjetje kot procesor plačilnega prometa, procesor bankomatskega in POS prometa ter ostalih storitev bančnega zaledja operira z ogromno količino zaupnih podatkov in informacij, ki jih je potrebno ustrezno ščititi. Združba je v preteklosti uspešno pridobilo certifikat PCI DSS in (del) ISO/IEC 20000 (katerega ogrodje je knjižnica ITIL), v prihodnosti pa bo morda deloma ali v celoti potrebovalo tudi certifikat ISO/IEC 27001. PCI DSS je standard, ki ga je predpisal konzorcij največjih ponudnik kartičnih produktov, kot so Visa, American Express, Mastercard [1]. V največji meri je namenjen ravno za certificiranje združb v sektorju industrije plačilnih kartic. Certifikat standarda ISO/IEC 20000 organizacije ISO je na zahtevo stranke podjetje pridobilo z implementacijo (dela) ogrodja ITIL. Oba omenjena standarda bom opisal in na dejanskih primerih prikazal implementacijo določenih (izbranih) zahtev kot odziv na prepoznane varnostne ranljivosti iz modelov ArchiMate. Ker se določena poglavja med standardi prepletajo in pokrivajo ista področja, lahko del zahtev enega standarda pokrijemo z drugim standardom. V magistrskem delu sem si kot cilj zadal preveriti in pokazati, kako bi lahko s pomočjo integracije in prekrivanja zahtev med standardi koristno uporabili zahteve, ki so v podjetju že udejanjene in s tem v praksi izkoristiti prednosti, ki bi jih taka integracija prinesla. Tako bom v magistrskem delu najprej opredelil koncepte varnosti, v modelirnem jeziku ArchiMate predstavil nekaj modelov problemov s področja varnosti, pregledal vse tri omenjene »velike« varnostne standarde, prikazal nekaj primerov implementacije varnostnih zahtev za spopad z varnostnimi ranljivostmi za standarda PCI DSS in ITIL, hkrati pa so predstavljeni tudi dokumenti, ki so plod vpeljave teh standardov v združbo. V nadaljevanju bom med sabo primerjal opisane standarde (PCI DSS – ISO/IEC 27001, ITIL – ISO/IEC 27001), za oba para primerjav pa bom poskusil najti zahteve, ki bi jih tako lahko koristno uporabili pri udejanjanju ISO/IEC 27001. Na koncu sledi še predlog modela implementacije ISO/IEC 27001 v podjetje na podlagi PCI DSS in ITIL.

1.1 Metode dela

V magistrskem delu bo uporabljeno znanje, pridobljeno pri podiplomskem magistrskem študiju Informacijski sistemi in odločanje na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Prav tako bo uporabljeno znanje in izkušnje, pridobljene pri delu na področju razvoja programske opreme, predvsem spletnih aplikacij in spletnih storitev, ter tudi implementaciji varnostnih standardov v združbo v podjetju Bankart. Znanje se bo poleg omenjenega črpalo predvsem iz tujih in domačih znanstvenih virov (nekaj člankov je citiranih tudi v Web of Science), strokovnih člankov, prispevkov na konferencah, standardov, ogrodij; pa tudi iz ostale literature, kot so knjige, interni viri podjetja, domači in tuji promocijski dokumenti, objave na spletnih straneh, blogih ipd.

Magistrsko delo bo zajemalo analizo prepleta standardov varovanja informacij v plačilnem prometu v naslednjih točkah:

- Proučeno bo teoretično ozadje informacijske varnosti v procesiranju plačilnega prometa
- Izbrani in proučeni bodo standardi in ogrodja, ki nudijo pomoč pri varovanju informacij
- Prikazana bo preslikava med zahtevami izbranih standardov
- Predlagan bo enostaven model implementacije enega standarda na podlagi ostalih
- V zaključku bodo oblikovani končni sklepi in ugotovitve

1.2 Primerjava in integracija standardov – trenutno stanje v znanstveni in strokovni literaturi

Za primerjavo standardov bodo uporabljeni in citirani članki in prispevki, ki že opisujejo sorodno tematiko. Člankov in literature s področja varnosti in tveganj je – zaradi pomena tematike pričakovano - zelo veliko. O tej obširni tematiki je napisanih veliko število knjig, priročnikov, znanstvenih, strokovnih ter konferenčnih člankov, pa tudi poljudnih blog zapisov ter dokumentov na internetu ipd.

V različnih znanstvenih revijah je veliko člankov o posameznih standardih, primerih implementacij teh standardov, predlogih izboljšav, zgodovinskem razvoju standardov in ogrodij.

Raziskava trenutnega stanja znanstvenih virov s področja primerjave in integracije standardov in ogrodij pa je pokazala, da znanstvenih člankov o tej temi ni tako v izobilju. Nekaj je strokovnih člankov in člankov z različnih konferenc s področja informacijske tehnologije. Več literature o primerjavi in integraciji posameznih standardov sem našel prosto dostopnih bodisi v obliki dokumentov na spletnih straneh ali v kot zapisov v obliki bloga tistih združb, katerih primarna dejavnost je pomoč pri implementaciji in certifikaciji standardov in dobrih praks v podjetja in gre (lahko) tudi za reklamno oziroma predstavitveno gradivo takih podjetij.

Vseeno sem uspel pridobiti nekaj literature, ki mi je bila v pomoč pri pisanju tega magistrskega dela.

V nadaljevanju predstavljam ključno oziroma temeljno literaturo (poleg standardov), ki jo bom uporabil v magistrskem delu. Morse in Raval [2] opisujeta kontekst varnosti v sektorju procesiranja plačilnega prometa, tekst pa je primerno izhodišče za razumevanje varnosti in varnostnih standardov v nadaljevanju. Teoretično ozadje vpeljave varnostnih standardov v združbe v svojem članku opisujeta Siponen in Willison [3], konsistenten pregled teorije informacijske varnosti pa nudi tudi knjiga Hintzbergena idr. z naslovom Foundations of Information Security Based on ISO 27001 and ISO 27002 [4]. V članku z naslovom Information Security Management System Standards: A Comparative Study of the Big Five [5] se avtorji lotijo sistematične primerjave petih glavnih standardov informacijske varnosti z več različnih vidikov in podajajo pregled njihovih lastnosti. Članek društva ISACA avtorja Tolge Mataracioglu [6] primerja standarda PCI DSS in ISO/IEC 27001. Podobno vsebino nudi tudi nekaj ostalih člankov (Lovrić [7], Blount [8]). Mubashir [9] se podobno loti primerjave ITIL in ISO/IEC 27001. Lovrić predlaga implementacijo PCI DSS standarda na podlagi ISO/IEC 27001, gre za integracijo v obratni smeri, kot jo v tem magistrskem delu predlagam sam. Navedena literatura bo torej koristila za študij primerjave in integracije standardov in jo bom uporabil kot osnovo za predlog modela implementacije ISO/IEC 27001 v združbo na podlagi PCI DSS in ITIL.

1.3 Struktura magistrskega dela

Magistrsko delo je vsebinsko razdeljeno na 7 poglavij. V 1. poglavju je uvod, opisane so metode dela in struktura magistrskega dela. V 2. poglavju je narejena predstavitev podjetja in dejavnosti. V 3. poglavju sledi teorija o konceptih varnosti in tveganjih ter upravljanjih tveganj. Prav tako je v 3. poglavju narejen pregled standardov s področja varnosti, v jeziku ArchiMate pa so podani primeri tveganj in ranljivosti, na podlagi katerih so v nadaljevanju opisani obravnavani standardi in ogrodja. Sledijo tri poglavja, v katerih so podani standardi in udejanjanje teh standardov na primerih iz v 1. poglavju opisane združbe. Tako je najprej opisan standard PCI DSS in primeri (4. poglavje), nato ogrodje ITIL in primeri (5. poglavje) ter standard ISO/IEC 27001 (6. poglavje). 7. poglavje je namenjeno primerjavi in integraciji omenjenih standardov in predlogu za udejanjenje ISO/IEC 27001 na podlagi ostalih standardov. 8. poglavje sestoji iz zaključka,v katerem so na kratko končni sklepi in ugotovitve, 9. poglavje je namenjeno pregledu virov in literature, 10. poglavje pa je dodatek z več preglednicami.

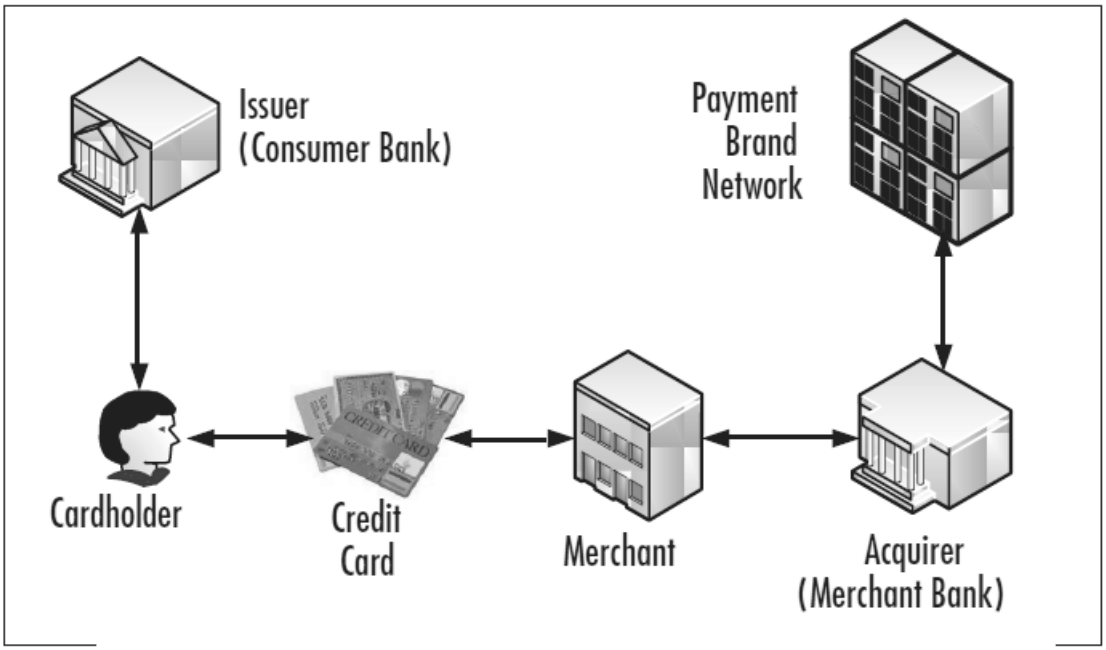
2 Podjetje in predstavitev dejavnosti

Namen tega poglavja je uvodoma podati jedrno dejavnost združbe, ki se v nadaljevanju nanaša na temo tega magistrskega dela in osnovno predstaviti omenjeno združbo z vidika poslovnih funkcij, procesov in pripadajočih podprocesov. Razumevanje dejavnosti sektorja plačilnih kartic, še posebej pa razumevanje v razdelkih 2.1 in 2.4 predstavljenega razmerja med dvema skupinama uporabnikov plačilnih kartic (trgovci in stranke), pa tudi pravnega in ekonomskega okolja, v katerem se procesiranje dogaja, je koristen predikat za razumevanje varnosti [2].

2.1 Dejavnost sektorja plačilnih kartic (*Payment Card Industry*)

Ena od možnih opredelitev dejavnosti sektorja plačilnih kartic pravi, da se industrija procesiranja plačilnih kartic nanaša na množico dejavnosti, povezanih z bankomati (*ATM – automated teller machine*), POS terminali (*POS – point of sale terminal*), kreditnimi, debetnimi, vrednostnimi, namenskimi plačilnimi ter darilnimi karticami ipd. [10]

Procesiranje plačilnih kartic je dejavnost, vpeta med dve medsebojno povezani tržišči. Kot je videti na sliki 1, so to na eni strani kupci, lastniki oziroma uporabniki (*cardholder*) plačilnih kartic (izdajateljska stran) in trgovci (*merchant*), ki te kartice sprejemajo (pridobitna stran). Obstajata dve vrsti sistemov plačilnih kartic, in sicer enoten sistem, pri katerem združba vzdržuje funkcije, ki zajemajo tako izdajo kartic potrošnikom kot tudi sprejem teh kartic na trgovski strani. Tak primer sta podjetji American Express in JCB (Diners); to sta finančni instituciji, ki sta isti tako na izdajateljski kot na pridobitni strani, sami torej tudi izdajata kartice. Obstaja še neenoten sistem, ki zajema več neodvisnih entitet, povezanih v omrežje, te neodvisne entitete pa tekmujejo za potrošnike (izdajateljska stran) in trgovce (pridobitna stran) [2] – primer takih podjetij sta Mastercard in Visa. V takem primeru imajo podjetja v ozadju finančne institucije, ki kartice izdajajo in finančne institucije, ki skrbijo za trgovce. Največkrat so te finančne institucije banke. Ker pa izdaja in procesiranje kartic ni tradicionalna dejavnost bank, v imenu slednjih to lahko opravljajo podjetja za procesiranje plačilnega prometa (kot v primeru podjetja, obravnavanega v tem magistrskem delu).



Slika 1 - Entitete, vpletene v industrijo in procesiranje plačilnih kartic
Vir: [1]

2.2 Podatki o imetnikih plačilnih kartic

Podatki, ki se nanašajo na transakcije plačilnega prometa, so shranjeni v finančnih ustanovah (banke oziroma še večkrat za njih združbe procesorji) plačilnega sistema [11]. Finančna institucija pridobitne strani (trgovca) zadržuje podatke vseh naročil, ki so nastala pri tem trgovcu, tudi podatek o številki računa tistega, ki je kupil oziroma naročil dobrine ali storitve tega trgovca [4]. Na drugi strani finančna institucija izdajateljske strani (lastnika plačilne kartice) zadržuje podatke, ki se nanašajo na lastnikove opravljene transakcije, na podlagi katerih mu lahko ponudijo mesečne izpiske prometa [11].

Iz različnih razlogov, vključno s samoumevnim varovanjem podatkov o računih strank, velikim tveganjem ugleda finančnih institucij, ki ne varujejo zaupnosti strank, tradicionalnega regulativnega nadzora v finančni industriji ter množice različnih posebnih predpisov so se finančne institucije zavezale k velikim investicijam v naložbe v informacijsko varnost za zaščito podatkov strank, predvsem seveda varovanju podatkov o lastnikih plačilnih kartic [11].

2.3 Plačilne kartice

Kot je videti iz slike 1, je osrednji pojem dejavnosti procesiranja plačilna kartica – plastični denar. Plačilne kartice so lahko različnih vrst:

- kreditne kartice (povezane z bančnim računom z odloženim plačilom),
- debetne kartice (povezane z bančnim računom z neposredno bremenitvijo),
- vrednostne kartice (kartice v fizični obliki ali obliki neke edinstvene šifre z določeno največjo vrednostjo)
- darilne kartice (kartice, na katerih je shranjena vrednost denarja, pri čemer obstajata dva tipa: darilne kartice, ki jih izda trgovec ali darilne kartice, ki jih je izda banka; ena od vrst darilnih kartic so tudi kartice zvestobe, ki se jih uporablja za identifikacijo kupca, pri čemer podjetje (največkrat trgovci) ponudijo tudi ugodnost za zvestobo kupca).

Glede na tehnologijo, se kartice delijo na kartice z magnetnim zapisom, pametne kartice in brezkontaktne kartice.

2.4 Avtorizacija plačil

Avtorizacija plačil se navadno izvede v dveh korakih: prvi korak je tok transakcije, drugi pa kliring in poravnava. Poenostavljen opis avtorizacije plačil povzemam po [12].

1. korak: Avtorizacija (*authorization*)

Transakcijski proces se začne, ko lastnik potisne svojo kartico v režo plačilnega terminala (POS terminala) ali pa vnese podatke kartice v spletno stran za nakup v e-trgovini. Trgovec, ki ima zakupljen terminal ali ki je lastnik spletne trgovine, zapiše tip kartice, številko kartice, datum poteka veljavnosti kartice in ostale kode. Te podatke in vrednost transakcije nato pošlje k pridobitelju, odgovornem za trgovčeve transakcije s plačilnimi karticami. Banka pridobiteljica nato pošlje podatke o transakciji izdajatelju kartice čez varovano omrežje plačilnega prometa. Izdajatelj kartice preveri status računa (glede na poslane podatke) v bazi in odgovoru pridobitelju, ki nato posreduje avtorizacijsko kodo terminalni napravi. V nekaterih primerih pridobitelj lahko transakcijo avtorizira neposredno, brez pošiljanja transakcijskih podatkov izdajatelju. Opisan postopek je seveda poenostavljen in je lahko drugačen za določene transakcije in v različnih državah.

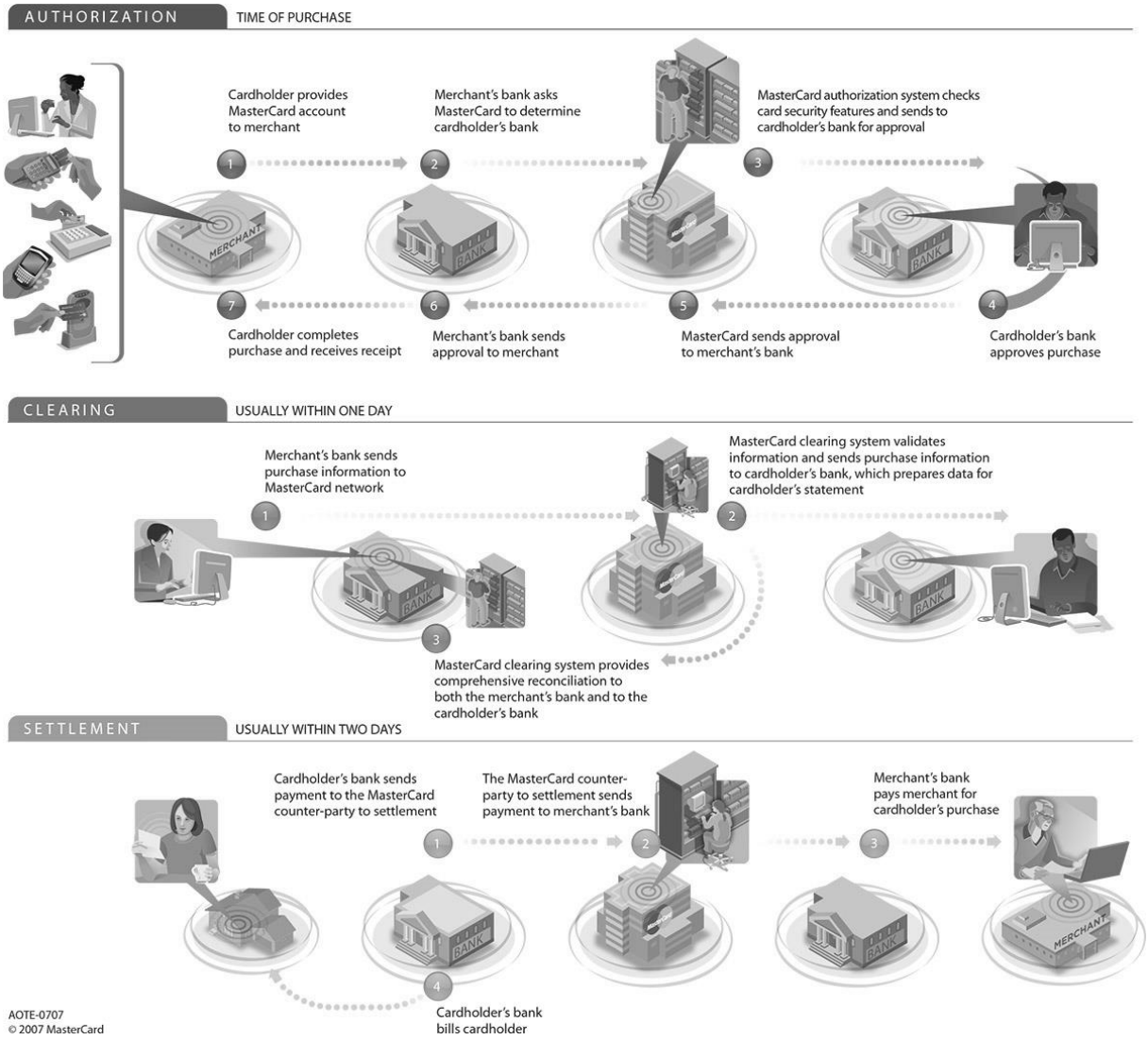
Sam transakcijski proces do sedaj še ne zajema dejanskih zaračunavanj, čeprav se na lokaciji trgovca s procesom prodaje nadaljuje. Z uspešno avtorizacijo gre namreč le za strinjanje izdajatelja kartice s poslom in poravnavo do pridobitne strani in trgovcem. Mali trgovci navadno pošljejo podatke o dnevnih transakcijah na koncu delovnega dne, medtem ko veliki to delajo »na liniji« (*on-line*), torej v realnem času za vsako transakcijo posebej. Proces zbiranja dolgov banke izdajateljice do banke

trgovca in povračila le-teh se lahko začnejo v trenutku, ko so podatki o transakciji poslani pridobitelju. Ta proces se imenuje kliring in poravnava, opisan v naslednji točki.

2. korak: Kliring in poravnava (*clearing, settlement*)

Ko pridobitelj pridobi transakcijske podrobnosti, podatke o njih le-ta pošlje ustreznemu plačilnemu omrežju (kot so Visa, Mastercard), od tam pa so ti podatki usmerjeni ustreznim izdajateljem kartic. Izdajatelj zaračuna lastniku – imetniku kreditne kartice, s katero je bila narejena transakcija, transakcijsko vsoto in pridobitelju preko omrežja nakaže ta sredstva, zmanjšana za provizijo oziroma pristojbino izdajatelja. Pridobitelj odšteje pristojbino za izdajatelja, omrežje (Mastercard, Visa, itd.) in za sebe ter tako nakaže preostanek sredstev na trgovčev račun. Ves ta cikel (zaračunavanje stranki, nakazilo trgovcu) se zgodi v 24 do 72 urah.

Opisan postopek avtorizacije plačil, kot ga predstavlja Mastercard, je viden na sliki 2.



Slika 2 - Avtorizacija transakcije kot jo predstavlja Mastercard
Vir: [13]

2.5 Osnovna predstavitev podjetja

Storitveno podjetje deluje na področju nudenja finančno-bančnih storitev. Podjetje skrbi za razvoj in upravljanje računalniške podpore, ki pri poslovanju s plačilnimi karticami, bančnimi avtomati, POS-terminali in drugimi sodobnimi tržnimi potmi zagotavlja ustrezno evidenco finančnega poslovanja posameznika. Ključna področja delovanja podjetja so naslednja [14]:

- Procesiranje kartičnega poslovanja
- Procesiranje bankomatskega poslovanja
- Procesiranje SEPA[1] kreditnih plačil
- Procesiranje SEPA direktnih obremenitev
- Procesiranje preko sistema E-račun

V nadaljevanju na kratko predstavljam vsako od zgornjih petih področij [14].

Kartično poslovanje temelji na čipni tehnologiji, podjetje pa procesira debetne oziroma plačilne kartice in tudi ostale kartične produkte, kot so Visa, Mastercard in Karanta.

Procesiranje bankomatskega poslovanja zajemajo storitve, kot so prenos sredstev med računi, avtomatski polog gotovine, elektronsko plačilo univerzalnih plačilnih nalogov (UPN), hitri dvig gotovine, dvig zneska po izbiri, izpis prometa po osebnem računu, vpogled v stanje na osebnem računu, sprememba osebnih (PIN) identifikacijskih števill, nakup GSM-kartic, vpogled v stanje na kreditnih karticah, naročilo za polog gotovine, naročilo za poravnavo plačilnih nalogov.

SEPA Infrastruktura za mala plačila (kratko SIMP) je sistem v katerem delujejo plačilni sistemi SEPA za obdelavo SEPA plačilnih instrumentov. Z vzpostavitvijo SIMP se je podjetje pridružilo ostalim evropskim podjetjem, ki zagotavljajo procesiranje kreditnih pa tudi debetnih plačil SEPA v enotnem standardu. Domača kreditna plačila SEPA (plačila znotraj države) se procesirajo med udeleženkami plačilnega sistema SEPA IKP, medtem ko se vsa čezmejna kreditna plačila SEPA procesirajo preko sistema SEPA EKP. Na področju direktnih obremenitev SEPA, se domača plačila procesirajo v okviru plačilnih sistemov SEPA IDD CORE in SEPA IDD B2B, vsa čezmejna plačila pa v okviru sistemov SEPA EDD CORE in SEPA EDD B2B.

Sistem E-račun, ki ga je razvilo podjetje, omogoča izmenjavo računov med pošiljatelji in prejemniki računov v elektronski obliki z uporabo elektronske banke. Z vzpostavitvijo enotnega sistema E-račun je komitentom bank in Upravi Republike Slovenije za javna plačila omogočena učinkovita izmenjava dokumentov v elektronski obliki. Vzpostavitev sistema E-račun pomeni ogromen prihranek pri manipulativnih in poštnih stroških, poenostavlja obstoječe procese izmenjave računov v podjetjih in prejemu pri potrošnikih. Z vzpostavitvijo sistema E-račun je podjetje svoje storitve procesiranja razširilo tudi na področje izmenjave računov v elektronski obliki.

Poslanstvo podjetja je opredeljeno kot [14]:

- Zagotoviti zanesljivost, varnost in stroškovno učinkovitost pri obdelavi transakcij z različnimi razlogi in bančnimi plačilnimi instrumenti.
- Zagotoviti skrben razvoj, gradnjo in vzdrževanje informacijskega okolja, s katerim se vsem strankam lahko omogoči nemoteno in kakovostno uporabo storitev.

[1] SEPA je kratica za Single Euro Payments Area, ki predstavlja enotno območje plačil v evrih – evro območja. SEPA je okolje, kjer lahko posamezniki, gospodarske družbe in drugi uporabniki plačilnih storitev v bankah izvajajo in prejemajo plačila v evrih, ne glede na to ali se takšno plačilo izvaja znotraj posamezne države ali med državami evro območja. Tovrstna plačila se izvršujejo pod enakimi osnovnimi pogoji, pravicami in obveznostmi ter poslovnimi običaji, ne glede na geografsko območje, državo nalogodajalca oziroma prejemnika plačila v okviru evroobmočja [15].

Vizija podjetja je usmerjena v iskanje novih priložnosti in izzivov v jugovzhodnem delu Evrope [14].

Cilji podjetja pa so s ponudbo visokokakovostnih in tehnološko najzahtevnejših storitev utrditi položaj vodilnega procesnega centra v državi in postati eden najkakovostnejših procesnih centrov v tem delu Evrope [14].

2.5.1 Poslovni procesi in poslovne funkcije podjetja

Z vidika informacijske tehnologije in vprašanja varnosti in tveganj ter primerjave standardov so v nadaljevanju opisane poslovne funkcije, ki iz informacijskega vidika orišejo dejavnost podjetja. Združba je identificirala pet visokonivojskih procesov, ki so opredeljene tudi v uradni dokumentaciji; ti so: procesiranje, podpora procesiranju, razvoj, upravljanje družbe in podporni procesi. Znotraj vsakega od naštetih funkcij so našteti poslovni procesi, pod njimi pa še podrobneje poslovni podprocesi. V naslednjem razdelku jih naštevam v celoti.

2.5.2 Visokonivojske poslovne funkcije podjetja

Glavne poslovne funkcije podjetja, ki jih identificira podjetje [16], so:

- Procesiranje
- Podpora procesiranju
- Razvoj
- Upravljanje družbe
- Podporni procesi

Jedrna funkcija podjetja je seveda procesiranje, ki je zadolžena za glavne poslovne procese podjetja, kot so kartično poslovanje, bankomatsko poslovanje, procesiranje podatkov izven Slovenije in SEPA instrumenti za mala plačila. Podpora procesiranju je funkcija, ki, kot že ime pove, skrbi za podporne funkcije in procese procesni funkciji, kot so storitveni servis (spremljava poslovanja, certifikati in ključi, klicni center), vzdrževanje IT infrastrukture in nadzor sistemov. Naslednja pomembna funkcija podjetja je razvoj, ki skrbi za zasnovo, razvoj in prodajo novih storitev in produktov. Upravljanje družbe je zadolženo za kakovostno vodenje in odločanje, skrb za varovanje in upravljanje informacij ter upravljanje nadzora in kontrole. Podporni procesi so zadnja velika poslovna funkcija in vključujejo finance, računovodstvo, splošne zadeve, servis, kadrovanje, izobraževanje, ipd. V naslednjem razdelku je zgornjih pet krovnih procesov razdeljenih na podprocese.

2.5.3 Procesi in pripadajoči podprocesi

V tem razdelku so strukturirano našteti procesi in njihovi (morebitni) podprocesi vsake od krovnih poslovnih funkcij, kot jih identificira podjetje samo [16]. Ti so:

- Procesiranje
    - Kartično poslovanje
        - Plačilni sistem poravnava kartic
        - Negotovinsko kartično poslovanje
    - Bankomatsko poslovanje
        - Plačilni sistem poravnava bankomatov
        - Negotovinsko bankomatsko poslovanje
    - Procesiranje izven Slovenije
    - SIMP
        - Plačilni sistem IKP
        - Sistem EKP
        - Sistem EDD B2B
        - Sistem EDD Core

        - Plačilni sistem IDD B2B
        - Plačilni sistem IDD Core
        - E-račun

- Podpora procesiranju
    - Storitveni servis
        - Spremljava poslovanja
        - Certifikati in ključi
        - Klicni center
    - Vzdrževanje IT infrastrukture
        - Vzdrževanje strojne in programske opreme
        - Vzdrževanje parametrov
        - Vzdrževanje komunikacij
    - Nadzor sistemov
- Razvoj
    - Razvoj in implementacija
        - Razvoj novih storitev
        - Implementacija novih strank, produktov
    - Prodaja
- Upravljanje družbe
    - Vodenje
    - Upravljanje in varovanje informacij
        - Fizično varovanje
        - Tehnično varovanje
    - Upravljanje nadzornih in kontrolnih funkcij
        - Notranje revidiranje
        - Upravljanje tveganj
        - Upravljanje neprekinjenega poslovanja
        - Zagotavljanje skladnosti
- Podporni procesi
    - Finance in računovodstvo
        - Finance
        - Računovodstvo
    - Upravljanje poslovnih prostorov in opreme
        - Prostori
        - Oprema
        - Vozni park
    - Promocija
    - Servis in inštalacije POS terminalov
    - Upravljanje s človeškimi viri
        - Kadrovanje
        - Izobraževanje

2.5.4 Razvoj in podfunkcije razvoja

Zaradi lastne vpetosti v razvoj programske opreme v podjetju sem za detajlnejšo predstavitev in modeliranje varnostnih tveganj ter podrobnejše predstavitve varnosti in varnostnih standardov izbral poslovno funkcijo razvoja poslovnih rešitev. Ta poslovna funkcija ni del uradne predstavitve funkcij

podjetja; identificiral sem jo sam, za potrebe tega magistrskega dela. Le-ta bi lahko bila ena od podfunkcij krovne funkcije razvoja. Ostale podfunkcije pa bi lahko bile:

- Razvoj in vzdrževanje bankomatskih in pos produktov
- Razvoj in vzdrževanje kartičnih produktov
- Razvoj plačilnih sistemov SEPA in E-Račun
- Implementacija novih strank
- Razvoj poslovnih rešitev

2.5.5 Poslovna funkcija razvoja poslovnih rešitev

Oddelek razvoja zalednih sistemov je edini popolnoma razvojni oddelek v podjetju; v njem je torej glavna poslovna funkcija razvoj programske opreme in uporabniških aplikacij. Med drugim tu poteka razvoj in vzdrževanje več spletnih storitev (*web services*)[2] in spletne aplikacije, ki jih podjetje nudi bankam, razvoj in izdelava poročil o bankomatskem in pos prometu za banke in trgovce, programska oprema za notranjo uporabo (klicni center in monitoring).

[2] Spletne storitve z uporabo protokola prenašajo sporočila napisana v jeziku XML. Uporabljajo standardne protokole spletne komunikacije, zato so neodvisne od operacijskega sistema in lahko povezujejo aplikacije, ki tečejo na različnih operacijskih sistemih, strojni opremi, različni programski opremi in podatkovnih bazah. Lahko tudi povezujejo več aplikacij iz različnih virov, da se ustvari vtis enotne storitve. Prav tako implementacija spletne storitve ni vidna zunaj spletne storitve [17].

3 Teoretično ozadje varnosti in tveganj

Uporaba sodobne tehnologije iz dneva v dan povečano pokriva večino vidikov našega življenja. Z napredkom tehnologije je čedalje več informacij na voljo prek medijev, kot je internet in s tem preprosteje dostopno za množice. Informacije se lahko štejejo za najpomembnejše sredstvo vsake moderne združbe [18]. Varovanje informacij mora zato biti ena od najpomembnejših nalog za vse združbe vseh velikosti ne glede na to, s čim se ukvarjajo. Še posebej pa to velja za podjetja, ki se ukvarjajo z zaupnimi podatki, kot so zdravstveni podatki ali finančni podatki. Združbam grozijo tveganja s področja informacijske varnosti, ki so posledica tako zlonamernih ali malomarnih dogodkov kot tudi neprimernih procesnih modelov, povezanih z avtorizacijami, dovoljenji dostopa in ločitvami nalog [19].

Po [1] največjo grožnjo za kršitve varnosti v združbah predstavljajo naslednje:

- Brezžična omrežja
- Manko mrežne segmentacije
- Izkoriščanje aplikacij na daljavo
- Dostopi zaposlenih znotraj podjetja

Vsako podjetje mora zato prepoznati, oceniti in analizirati tveganja s področja informacijske varnosti. Če niso sposobna prepoznati tveganj v povezavi s tehnologijo, ki jo uporabljajo, z ljudmi, ki jih zaposlujejo ali z okoljem, v katerem delujejo, lahko pride do nepredvidenih posledic, ki lahko privedejo do večje škode pri poslovanju [20]. Združbe se morajo zavedati potrebe, da namenijo več sredstev za zavarovanje informacijskih sredstev, pri čemer mora biti informacijska varnost glavna skrb tako vladnih združb kot podjetij, ki nastopajo na trgu [21].

Po [4] je za informacijsko varnost posebnega pomena najti ravnovesje med več vidiki. Ti vidiki so:

- Kvalitativne zahteve, ki jih morda združba ima glede informacij
- Tveganja, povezana s temi kvalitativnimi zahtevami
- Protiukrepi, potrebni za ublažitev teh tveganj
- Zagotavljanje neprekinjenega poslovanja v primeru katastrofalnih dogodkov
- Kdaj in v katerih primerih poročati o incidentih zunaj združbe

Navodila za upravljanje varovanja informacij poskušajo ponuditi najboljše prakse. Združbe lahko z uporabo navodil izkažejo svojo pripravljenost uporabe teh najboljših varnostnih praks in tako zaprosijo za certifikacijo, akreditacijo ali testiranje klasifikacije varnostne zrelosti glede na njihovo kompatibilnost na skupek pravil in praks [3]. V informacijski tehnologiji je na voljo veliko število pristopov različnih kategorij, kot so ogrodja, standardi, regulative, metodologije ipd[3].

3.1 Koncepti varnosti

V nadaljevanju razdelka so opisani osnovni pojmi, ki opredeljujejo informacijsko varnost in ki jih uporabljam skozi celotno magistrsko delo.

3.1.1 Osnovni pojmi

3.1.1.1 *Tveganje*

Tveganje je verjetnost, da se bo zgodila določena škoda, pri čemer gre za možnost, da bo izid dogodka drugačen od predvidenega. Če je na primer na komunikacijskem nivoju še omogočena enkripcija prometa z verzijo SSL verzije 2, obstaja večja verjetnost, da bo napadalec imel možnost nepooblaščeno prisluškovati prometu, kot če bi bil omogočen le najbolj varen protokol (TLS 1.2). Prav tako na primer obstaja večja verjetnost, da bo razvijalec programske opreme ne namenoma

[3] V nadaljevanju bom pisal o standardih, ogrodjih, regulativah, uredbah, principih ipd. Največkrat bom uporabil termin standard, čeprav se z uporabo tega termina (lahko) implicitno razume tudi druge naštete termine.

naredil napako pri delu s podatki v podatkovni bazi (npr. pobrisal transakcijske podatke), če ne uporablja ali podjetje sploh ne razpolaga s postopki in dokumentacijo za operativna navodila, kot je delo s podatkovno bazo. Tveganje povezuje ranljivost, grožnje in verjetnost izkoriščanja k nastalemu poslovnemu vplivu [4].

Lahko pa tveganje opredelimo tudi matematično. Tveganje (*risk*) ima dva osnovna atributa: verjetnost (*probability* P) in vpliv (*impact* I), pri čemer verjetnost opisuje možnost, da se nek dogodek zgodi, vpliv pa pomeni posledice tega dogodka, ko se le-ta zgodi. Tveganje (Rx) lahko torej tako tudi matematično opredelimo kot funkcijo dveh atributov: Rx=f(Px,Ix) [22].

3.1.1.2 *Grožnja*

Grožnja je možen vzrok za neželen incident. Tak incident lahko naredi škodo na sistemu ali škodo družbi. Entiteto, ki izkoristi ranljivost, imenujemo agent grožnje. Agent grožnje je lahko tako subjekt kot objekt. Subjekt je lahko npr. napadalec, ki vdira s pomočjo spletnih ranljivosti, na drugi strani pa je grožnjo lahko predstavlja objekt, kot so vremenske nevšečnosti ali računalniški proces, ki dostopa do podatkov na način, ki je v nasprotju varnostni politiki podjetja [4]. Grožnja je po drugi opredelitvi lahko tudi možen vzrok nezaželenega vpliva na nekem sistemu [22].

3.1.1.3 *Ranljivost*

Ranljivost je slabost (šibkost ali napaka) določenega sredstva (v ukrepih v sistemu, sami arhitekturi sistema ali njegovi izvedbi, notranjih kontrolah ter ostalih vzrokih), s katerim razpolaga podjetje, ki ga lahko določena grožnja izkoristi. Primeri ranljivosti so operacijski sistem in aplikacije brez zadnjih varnostnih popravkov, nepotrebni odprti porti v požarni pregradi, nezavarovana sistemska soba ipd [4], [22].

3.1.1.4 *Izpostavljenost*

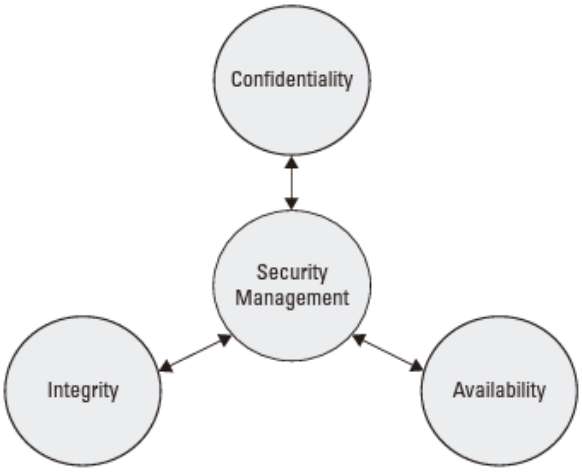
Izpostavljenost je stanje, pri katerem je nekdo ali nekaj izpostavljen izgubam s strani grozečega subjekta ali objekta. Ranljivost na primer izpostavi združbo pred možno škodo. Primer izpostavljenosti je npr. kadar združba nima politike o upravljanju z gesli in je tako možno uporabljati dovolj enostavna gesla s tem pa je izpostavljena možnosti, da so enostavna gesla prestrežena in s tem napadalec dobi možnost dostopa do občutljivih podatkov, do katerih sicer ni avtoriziran. Drug primer je, kadar združba ne opravlja proaktivnega nadzora pri preventivi pred požari in s tem izpostavlja nevarnost uničujočega požara [4].

3.1.1.5 *Zaščita in protiukrep*

Protiukrep je dejanje, s katerim zmanjšamo oziroma odbijemo možno tveganje. To je nastavitev programske opreme, naprava, procedura ali kakšno drugo sredstvo, s katerim odstranimo ranljivost ali zmanjšamo možnost, da bi nek agent grožnje izkoristil ranljivost. Primeri protiukrepov so uporaba in upravljanje močnih gesel, varovanje, mehanizmi kontrole dostopa, ozaveščanje zaposlenih o pomenu varnosti ipd. [4]

3.1.2 CIA trikotnik

Pojmi zaupnost (*Confidentiality*), celovitost (*Integrity*) in razpoložljivost (*Availiability*) predstavljajo tako imenovani CIA trikotnik, ki je smatran kot osnovni princip programov in ogrodij varnosti in tudi upravljanja varnosti. Ker so informacije lahko najdragocenejše sredstvo določene združbe, je po mnenju de Oliveira Albuquerque in ostalih [18], varovanje teh informacij vključuje ravno ščitenje pojmov CIA trikotnika, zaupnosti, celovitosti in razpoložljivosti. Vedno lahko, ko pomislimo na varnost in implementacijo varnosti, uporabimo koncepte, ki jih predstavlja CIA [4].



Slika 3 - CIA trikotnik
Vir: [4]

Zaupnost je pojem, ki opredeljuje, kaj lahko kdo vidi, hkrati pa varuje podatke pred neavtoriziranim dostopom. Zaupnost mora biti zagotovljena za rezidirane podatke, ko se le-ti prenašajo, pa tudi, ko so v uporabi. Koncepti, ki se v informacijski tehnologiji povezujejo z zaupnostjo, so kriptiranje, avtentikacija, avtorizacija, kontrola dostopa, klasifikacija podatkov ipd [4].

Integriteta ali celovitost se nanaša na stanje, pri katerem so podatki v vsakem trenutku pravilni in konsistentni, pri čemer morajo biti spremembe na teh podatkih vedno avtorizirane, torej je spremembo naredil nekdo, ki ima za to ustrezne pravice. Takoj ko se zgodi neavtorizirana sprememba podatkov, gre za kršitev varnosti [4].

Razpoložljivost opredeljuje stanje, pri kateri je informacija, ki ga nekdo, ki je avtoriziran za dostop do nje, v vsakem trenutku in na vnaprej opredeljenem mestu lahko pridobi. Po [4] so lastnosti razpoložljivosti naslednje: pravočasnost (informacija je dostopna v vsakem trenutku), kontinuiteta (čeprav pride do okvare dela sistema, uporabnik lahko nadaljuje z delom) in robustnost (na razpolago je dovolj kapacitete za delo) [4].

3.2 Upravljanje tveganj

Upravljanje tveganj vključuje identifikacijo in implementacijo učinkovitih varnostnih kontrol, s katerimi omilimo, nadzorujemo in rešimo tveganja združbe [23].

Informacijska varnost je del upravljanja tveganj, naloga je upravljanje tveganj, povezanih z razkrivanjem informacij [18].

Tveganja, ki ogrožajo varnost informacijskih in računalniških virov morajo biti ocenjena in upravljana na pravilen način in s potrebnimi varnostnimi kontrolami, ki morajo biti udejanjene in upravljanje učinkovito [24].

Upravljanje tveganj informacijske tehnologije je umetnost prepoznave obstoja groženj in vpliva posledic na vire ter uporaba spreminjajočih se dejavnikov na stroškovno učinkovit način, da (morebitne) škodljive posledice ostanejo znotraj meja [18].

Rot [24] je v literaturi našel štiri glavne komponente upravljanja tveganj, in sicer so to:

- Prepoznavanje tveganja: v tej prvi fazi upravljanja tveganj naj bi združbe čim bolj zgodaj ugotovile morebitne realizacije tako zunanjih kot notranjih varnostnih groženj za celoten informacijski sistem. Prepoznavanje je proces iskanja, opisovanja, dokumentiranja in

komuniciranja o tveganjih, preden le-ti nastanejo problem, ki bi lahko združbo resno ogrozile [24].

- Analiza tveganja: Označuje se jo kot najpomembnejšo in ključno fazo upravljanja varnosti in posledično upravljanja tveganj. Z njo se ocenjuje tveganja, ki morajo biti nadzorovana, minimizirana in/ali sprejeta. Cilji analize so prepoznati sredstva in njihovo vrednost za združbo, določiti ranljivosti in grožnje, določiti tveganja, če se grožnje udejanjijo in motijo operativne postopke ter določiti razmerje med stroški morebitnega incidenta in stroški varnostnega ukrepa. Za analizo se uporablja več metodologij oziroma pristopov in sicer kvantitativen, kvalitativen in hibridni pristop. Hibridni je seveda mešanica kvantitativnega in kvalitativnega pristopa. Pri kvantitativni metodologiji je ocena tveganja izračunana na podlagi numeričnih vrednosti in cilja na izračun finančne izgube in verjetnosti, da bi grožnja resnično postala incident oziroma problem. Pri tej metodologiji ima vsak element pri vseh operativnih postopkih svojo vrednost. Te vrednosti so lahko sestavljene iz stroškov varnostnih ukrepov, kot tudi vrednosti premoženja samega, vključno s postavkami, kot so stavbe, strojna in programska oprema, informacije in vpliv na poslovanje. Upoštevati pa je potrebno tudi elemente, kot so čas do pojava grožnje, učinkovitost varnostnih ukrepov in tveganje, da se bo določena ranljivost dala izkoriščati. Z upoštevanjem zgoraj naštetega se pri tej metodi lahko dobi ocenjeno celotno finančno tveganje in se na podlagi tega lahko določijo ustrezni ukrepi. Stroški teh ukrepov ne smejo presegati vrednosti ocenjenih elementov in tveganja. Pri kvalitativni metodologiji elementi in izgube ne dobijo numeričnih vrednosti, pač pa gre za pregled različnih scenarijev možnosti tveganj z oceno resnosti groženj in veljavnosti morebitnih protiukrepov. Tehnike tovrstne metodologije so sodbe, praksa, intuicija in izkušnje, primeri pa *brainstorming*, *storyboarding*, fokusne skupine, ankete, vprašalniki, seznami, intervjuji ipd. Navadno se formira ekipo za analizo tveganj, ki imajo znanje in izkušnje o grožnjah; ti potem preigrajo različne scenarije in ocenijo tveganja [24] [4].
- Ukrepi za zmanjševanje tveganja: V tej fazi naj bi združbe udejanjile ukrepe za zmanjšanje tveganj za celotno informacijsko okolje. Na voljo je več različnih varnostnih ukrepov za več različnih varnostnih tveganj, razdeljenih v tri skupine. Prva skupina je administrativni nadzor, kot so odobrene politike, postopki, navodila, standardi. Druga skupina je tehnični nadzor, to je uporaba programske opreme za spremljanje in nadzor dostopov do informacijskega in računalniškega sistema (gesla, enkripcija, požarne pregrade, seznami dostopov ipd.). Tretja skupina pa je fizični nadzor okolja (vrata, zaklepanje, klimatizacija, požarni alarmi, varnostniki ipd.) [24]
- Spremljanje tveganja: Zadnja faza je aktivno spremljanje tveganja, aktivno spremljanje skrbi, da so protiukrepi v IT okolju ustrezno udejanjeni in aplicirani.

3.2.1 Ocena varnostnih tveganj

Svetovalec za upravljanje Peter Drucker je nekoč dejal [25]: »Če ne morete izmeriti, ne morete upravljati«. Iz njegovih besed izhaja, kar naj bi bilo tudi splošno znano: prvi korak pri varovanju informacij v vsaki združbi mora biti ocena varnostnega tveganja opreme in postopkov, ki se uporabljajo za zbiranje, procesiranje, hranjenje in distribucijo informacij.

Informacijsko varnost je moč zagotoviti z udejanjenjem nabora kontrol, kot so politike, procesi, procedure, organizacijske strukture, funkcije programske in strojne opreme ipd. Kontrole morajo biti kreirane, udejanjene, nadzorovane, pregledane in kontinuirano izboljšane skupaj z ostalimi procesi upravljanja poslovanja [4].

Varovanje informacij se največkrat dojema kot tehnološki problem, kar je preozko gledanje [26]: pravzaprav gre za problem upravljanja. Zato kot takšno potrebuje celovit pristop v obliki ukrepov, postopkov, standardov in politik.

3.2.2 Upravljanje varovanja informacij

Združbe bi morale v skladu z načelom PDCA (*plan-do-check-act*), ki zapoveduje štiri faze upravljanje varovanja informacij (vzpostavitev sistema, implementacija in delovanje, nadzor in pregled, vzdrževanje in izboljševanje) upravljati s sistemom za upravljanje varovanja informacij v kontekstu poslovnih aktivnosti in povezanih tveganj [4].

V osnovi je informacijska varnost proces, namenjen identifikaciji tveganj in zmanjševanju njihovih učinkov na karseda nizek nivo. Ta proces naj bi bil iterativnega značaja in naj bi potreboval sistem za upravljanje varovanja informacij (SUVI) (angl. *Information security management system – ISMS*). SUVI naj bi po opredelitvi [27] kot vhod vzel varnostne zahteve in pričakovanja združbe, na izhod pa ponudil rezultate informacijske varnosti, ki bi naslavljali omenjene zahteve in pričakovanja.

Združbe bi morale implementirati sistem za upravljanje varovanja informacij (SUVI), ki sestoji iz množice politik, ki jih združba opredeli, sestavi, razvije in vzdržuje in se nanašajo na strojne in programske računalniške vire [5].

Disterer [28] trdi, da učinkovit SUVI prispeva k zmanjšanju tveganj in zaščiti združbe pred kršitvami varnosti.

Iz metodološkega stališča je po ENISA[4] za razvoj SUVI potrebnih šest korakov [29]:

- opredeliti varnostno politiko
- opredeliti obseg, ki ga bo pokrival SUVI
- oceniti tveganja (kot del upravljanja s tveganji)
- upravljati tveganja
- izbrati primerne kontrole
- podati izjavo o uporabnosti

Alfantookh [2 v [5]] je opredelil 11 kontrol oziroma nadzornih korakov, poimenoval jih je 11EC, ki naj bi jih združbe udejanjile kot merila, ki jih uteleša SUVI:

1. Politika informacijske varnosti: opredeljuje, kako združba izraža namero zagotavljanja informacijske varnosti, podaja navodila vodstvu in zaposlenim o tej temi in obvešča ostale deležnike
2. Upravljanje komunikacij in delovanja: kako je v združbi opredeljena politika varnosti z namenom zmanjšanja varnostnih tveganj na podlagi operativnih postopkov, kontrol in dobro opredeljenih odgovornosti
3. Nadzor dostopa: določen sistem, ki subjektom omogoča nadzorovati dostop do virov v določenem fizičnem ali informacijskem okolju
4. Pridobitev, razvoj in vzdrževanje informacijskega sistema: integriran proces, ki opredeljuje meje ter tehnične informacijske sisteme od pridobitve, razvoja in do vzdrževanja informacijskih sistemov
5. Organizacija informacijske varnosti: struktura za implementacijo informacijske varnosti v lasti združbe; sestavljena je iz zavedanja vodstva (*managementa*) o informacijski varnosti, koordinacije informacijske varnosti ter odobritvenih procesov
6. Upravljanje sredstev: sloni na zamisli, da je pomembno identificirati, slediti, klasificirati in dodeliti lastništvo vsem najpomembnejšim sredstvom s ciljem zagotoviti njihovo učinkovito zaščito
7. Upravljanje incidentov s področja informacijske varnosti: vsebuje identifikacijo virov za upravljanje z incidenti. V primeru dobro zastavljenega upravljanja z incidenti se lahko prepreči nastanek novih incidentov

[4] European Network and Information Security Agency (ENISA)

8. Upravljanje neprekinjenega poslovanja: program za zagotavljanje neprekinjenega poslovanja v primeru izrednih dogodkov in/ali razmer. Načrti podajajo pripravljenost združbe za hitro vzpostavitev in obnovo poslovanja po ali med izrednimi razmerami, minimizira vpliv takih dogodkov in v takih primerih zagotavlja sredstva
9. Varnost človeških virov: vsi zaposleni, pa tudi pogodbeniki in ostali uporabniki, so kvalificirani za svoje delo in se zavedajo odgovornosti in dolžnosti, ki jih imajo pri opravljanju dela; Pomembno je, da so po končanju takšnega ali drugačnega delovnega razmerja odstranjeni vsi dostopi
10. Fizična varnost in varnost okolja: fizično okolje (zgradba, prostori, sobe, ostali sistemi), ki ga združba uporablja pri svojem poslovanju, je za preprečitev škode ali nedovoljenih dostopov ustrezno fizično zaščiteno
11. Skladnost: skladnost, razdeljena na dva dela; prvi del so zakoni, regulacije in pogodbene zahteve, drugi del pa skladnost s politiko informacijske varnosti, standardi in procesi

Glavni namen SUVI je po [30] ponuditi upravljanje z zaupanjem, da je varnost informacij združbe ustrezno upravljanja, kot je to zahtevano s strani vodstva in regulativnih zahtev (državni zakoni). Poleg tega naj bi SUVI zbiral dokaze, da združba skrbi za odgovorno poslovanje z upoštevanjem informacijske varnosti. Vodstvo je tisto, ki naj spodbuja k uveljavljanju SUVI in ga v združbi »prodaja, hkrati pa naj zaposleni sprejmejo, da je SUVI koristen za vse«. Vsak naj sprejme delček lastništva za uspeh, le tako bo SUVI resnično uporaben in učinkovit [30].

3.3 Standardi in ogrodja s področja informacijske varnosti

Pojav novih in razvoj obstoječih standardov se zgodi z razvojem podrobnih opisov posameznih značilnosti proizvoda ali storitve, ki jih podajajo strokovnjaki iz podjetij in znanstvenih ustanov. Predstavljajo konsenz o značilnostih, kot so kakovost, varnost in zanesljivost; s tem vedenjem se bodo ti izdelki in storitve za daljše časovno obdobje še naprej uporabljali. Zato so dokumentirani in objavljeni. Cilj razvoja standardov je podpora posameznikom in podjetjem pri naročanju izdelkov in storitev. Ponudniki izdelkov in storitev lahko povečajo svoj ugled, ki jih certificirajo skladno s standardi [28].

Da se združbe lahko soočajo z izzivi ocenjevanja in upravljanja tveganj, morajo vpeljati (tudi) mednarodno priznana ogrodja in dobre prakse v podjetje [20]. Splošno so standardi mišljeni kot doprinos k uniformnosti, ki pomaga pri razumevanju in upravljanju določenih področij [20]. Gonila k vpeljavi standardov so lahko različnega izvora. Iz poslovnega vidika se standarde vpeljuje zaradi zahtev poslovanja ali različnih regulativ in mandatov skladnosti. Vzpostavitev ustreznega korporativnega upravljanja, povečana zavest o tveganjih in tekmovalnost z drugimi podjetji, so še nekateri vzgibi za vpeljavo standardov. Tržni vidik in nastopanje na trgu je naslednja skupina razlogov za vpeljavo standardov, saj se od nekaterih združb pričakuje, da uvedejo določena ogrodja in dobre prakse, prav tako pa gre lahko tudi za ugled združbe v poslovnem okolju. Po [20] so ključni razlogi za vpeljavo standardov ponuditi zaupanje poslovnim partnerjem, strankam in ostalim deležnikom, zmanjšanje odgovornosti zaradi neizvedenih ali slabo izvedenih politik in procedur v podjetju, dobiti višje lastništvo upravljanja ter vzpostaviti mehanizem za merjenje uspešnosti varnostnih kontrol. Največje poslovno gonilo pa je prekritje pomanjkanja izkušenj na določenih področjih, kjer združba ni zmožna sama ustvariti standardov in dobrih praks na podlagi kompetenc zaposlenih [20].

3.3.1 Pregled standardov in ogrodij

Na področju informacijske varnosti so bili razviti različni standardi, ki naj bi v združbah poleg izboljšanja same varnosti omogočili tudi njen enostavnejši razvoj. Najbolj razširjeni SUVI standardi so naslednji:

- ISO/IEC 27001
- BS 7799

- PCI DSS
- ITIL / ISO/IEC 20000
- COBIT

Standardi, obravnavani v tem magistrskem delu (PCI DSS, ISO/IEC 20000, ISO/IEC 27001), so po [5] trije izmed peterice tako imenovanih velikih SUVI standardov, ki naj bi predstavljali merilo upravljanja informacijske varnosti.

3.3.2 Postopek certificiranja

Da združba dokaže svojo prilagojenost procesov zahtevam, ki jih določa nek standard ali ogrodje, mora s postopkom certificiranja pridobiti certifikat. Za večino standardov je omenjeni postopek certifikacije zelo podoben in po navadi poteka v treh fazah, in sicer [31]:

1. Priprave združbe na certifikacijo. V tej pripravljalni fazi združba razvije in udejanji svoj sistem upravljanja varovanja informacij (SUVI), vzpostavljen sistem integrira v svoje vsakodnevne poslovne procese in aktivnosti, usposobi svoje zaposlene in izvaja proces vzdrževanja.
2. Izvedba presoje o ustreznosti vzpostavljenega sistema upravljanja informacij oz. drugega ogrodja. V tej fazi akreditiran certifikacijski organ izvede presojo SUVI, ki ga je združba implementirala v prvi fazi. Ta faza se deli na dva dela. V prvem delu se preveri vzpostavljenost in dokumentiranost SUVI in zajema preglede varnostne politike in ciljev, obsega sistema, podpornih postopkov in kontrol ter poročilo o oceni tveganja, vpeljane programe in ukrepe za zniževanje tveganj in nazadnje izjavo o primernosti. V drugem delu se presoja izvajanje in učinkovitost sistema vodenja varovanja informacij, izpolnjevanje zahtev določenega standarda, zakonskih zahtev in zahtev deležnikov. Certifikate se podeljuje za določeno obdobje, potem pa jih je potrebno obnavljati oziroma je potrebna ponovna certifikacija in ravno to je predmet naslednje faze.
3. Kontinuirano vzdrževanje in izboljševanje (podaljševanje certifikacije). V tej fazi certifikacijski organi na določeno časovno obdobje obišče združbo in preveri ali združba še izpolnjuje predpisane zahteve.

3.4 Prikaz varnostnih tveganj v modelirnem jeziku ArchiMate

3.4.1 Namen uporabe ArchiMate v magistrskem delu

Z željo na primerih prikazati varnostna tveganja iz konkretnih in dejanskih problemov varnosti, s katerimi se soočamo pri poslovanju podjetja, sem poiskal ustrezen modelirni jezik in koncepte, na podlagi katerih bom v nadaljevanju predstavil varnostne standarde in preplet med njimi.

Ker so tveganja, povezana z informacijsko tehnologijo v veliki meri povezana z združbo in ne morejo ostati v osami, prepuščena le strokovnjakom s področja IT-ja kot je bilo to do nedavnega, so ta tveganja del t.i. upravljanja tveganj in varnosti na nivoju združbe (*Enterprise Risk and Security Management – ERM*). ERM združuje metode in tehnike, ki jih organizacija uporablja za upravljanje vseh vrst tveganj [32]. Kot je opredeljeno v nadaljevanju v razdelku 3.4.2, je ArchiMate jezik, ki se uporablja za modeliranje arhitekture s celostnim pregledom načrta in strukture podjetij (*Enterprise Architecture – EA*) in je kot tak primeren tudi za umestitev konceptov tveganja in varnosti.

3.4.2 Opredelitev ArchiMate

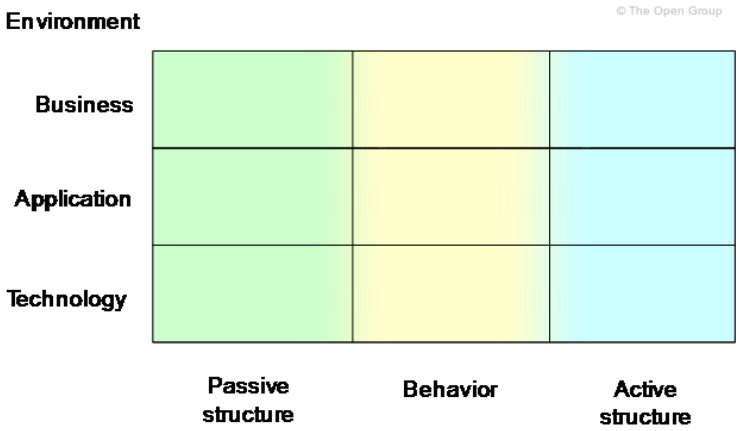
ArchiMate je po uradni opredelitvi [33] tehnični standard združbe The Open Group[5], ki je odprt in neodvisen modelirni jezik za arhitekturo podjetij (*Enterprise Architecture –EA*)[6]. Modelirni jezik, ki ga podpira več podjetij, omogoča arhitektom opisati, analizirati in vizualizirati razmerja znotraj in med različnimi poslovnimi domenami na nedvoumen načni.

V nadaljevanju uradna opredelitev opisuje primerjavo z arhitekturo v gradbeništvu: klasična arhitektura opisuje različne vidike konstrukcije, gradnje in uporabe stavbe, primerljivo tudi ArchiMate ponuja pregled nad konstrukcijo in uporabo poslovnih procesov, organizacijskih struktur, informacijskih tokov, IT sistemov in tehnične infrastrukture. Tak vpogled namreč lahko pomaga deležnikom načrtovati, oceniti in ukrepati o posledicah odločitev in sprememb v in med različnimi omenjanimi poslovnimi domenami.

ArchiMate je bil razvit v želji zagotoviti enoten nabor diagramov, ki opisujejo poslovne arhitekture. Kot opisuje The Open Group [34], je ArchiMate lahek in prilagodljiv z (vsaj) dveh vidikov:

- Ogrodje je preprosto, vendar dovolj močno, da zagotovi dobro strukturirani mehanizem za arhitekturo domene, plasti (poslovno, aplikacijsko, tehnološko) in različne vidike.
- Jezik vključuje koncepte storitveno orientirane paradigme, ki promovirajo nove organizacijske principe v smislu storitev za združbe z daljnosežnimi posledicami za arhitekturo podjetij.

Po Lankhorstu [35] ArchiMate nudi doslej najbolj celovit integriran pristop za izgradnjo, predstavitev in vzdrževanje arhitekture poslovnih sistemov, pri čemer je glavni cilj integracija arhitekturnih domen.



Slika 4 - Ogrodje ArchiMate
Vir: [33]

[5] The Open Group je neodvisno združenje, ki izvaja nekaj različnih programov za doseganja določenih poslovnih ciljev preko standardov informacijske tehnologije. Nastala je leta 1996 z združitvijo dveh združb, X/Open in The Open Foundation. Konzorcij The Open Group združuje več sto članov, med njimi so najvidnejša podjetja kot so Capgemini, Hewlett Packard Enterprise, Huawei Technologies, Co. Ltd, IBM, Oracle Corporation, Philips itd. Omenjeni programi oziroma storitve, ki jih opravljajo, so certificiranje, izvajanje forumov in delovnih skupin za izmenjavo mnenj in izkušenj ter dobrih praks, prirejanje dogodkov, nudenje storitev in izdajanje publikacij.

[6] Po opredelitvi [20] je arhitektura združbe (*Enterprise architecture - EA*) dobro opredeljen skupek praks in načel za izvedbo analize podjetij, oblikovanje, načrtovanje in implementacijo, pri kateri je v vsakem trenutku pomemben celosten pristop s katerim lahko podjetje uspešno razvija in izvaja svojo strategijo. Prakse vodijo združbe skozi poslovne, informacijske, procesne in tehnološke spremembe, potrebne za izvajanje strategij.

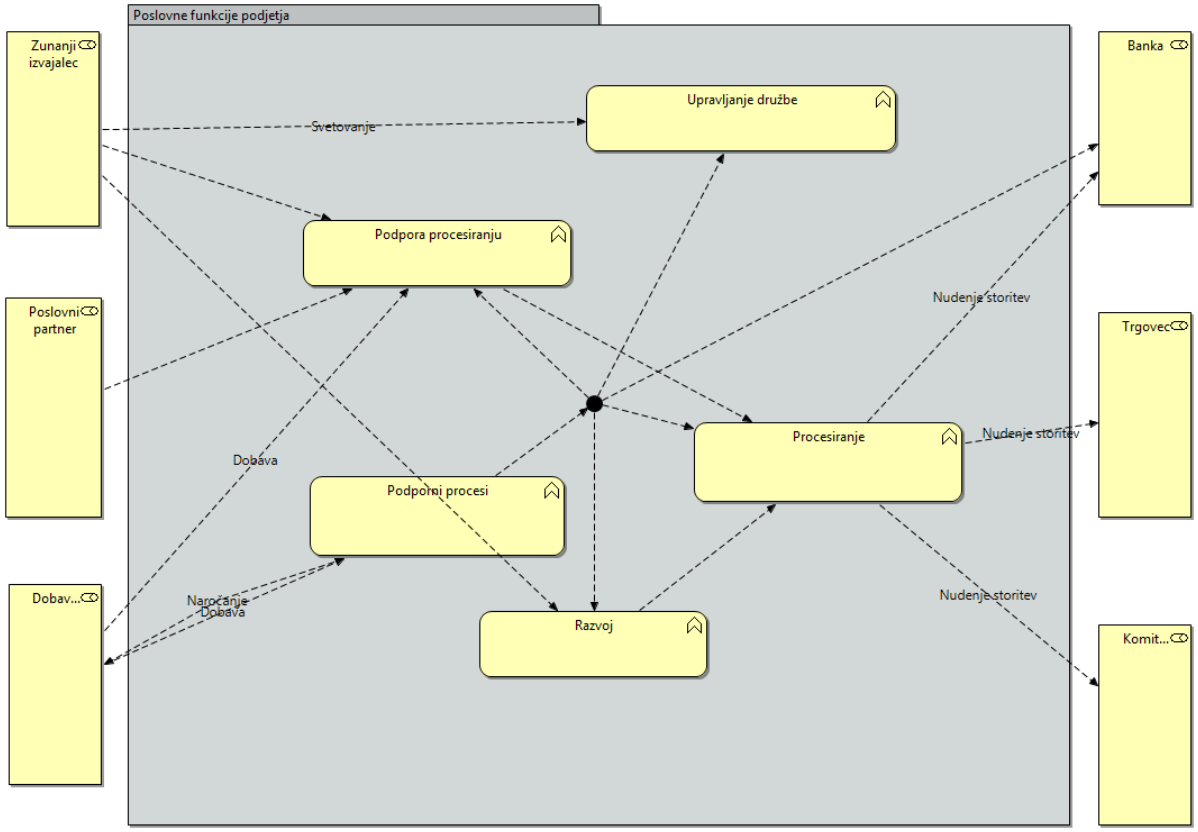
Na sliki 4 je prikazano ogrodje jezika ArchiMate, katerega ključni koncept je storitev in ki opredeljuje tri arhitekturne plasti: poslovno (*business*), aplikacijsko (*application*) in tehnološko (*technology*). Poslovni nivo ponuja izdelke in storitve, ki jih preko poslovnih procesov s strani poslovnih akterjev podjetje ponuja strankam. Aplikacijski nivo je podporni nivo poslovnega nivoja, slednji ponuja aplikacijske storitve, implementirane v programskih rešitvah. Tehnološki nivo ponuja infrastrukturne storitve, ki so potrebne za delovanje aplikacij, te infrastrukturne storitve pa omogočajo računalniška in komunikacijska oprema ter sistemska programska oprema [33]. Navpično ogrodje opredeljuje tri vidike, in sicer pasivno strukturo, obnašanje in aktivno strukturo.

Poleg tega ogrodje ponuja še nekaj ostalih konceptov, kot so cilji, načela in zahteve (*goals, principles and requirements*), tveganje in varnost (*risk and security*), upravljanje (*governance*), politike in poslovna pravila (*policies and business rules*), stroški (*costs*), uspešnost (*performance*), pravočasnost (*timing*) ter načrtovanje in razvoj (*planning and evolution*).

Glavne aktivnosti modeliranja so določitev namena, obsega in poudarka [36]. Nadalje je potrebno pri modeliranju izbrati zorni kot, s katerim naredimo predstavitev [36] posameznega modela. ArchiMate pozna naslednje načrtovalske zorne kote: organizacijska struktura, poslovna funkcija, poslovni proces, informacijska struktura, struktura aplikacij, obnašanje aplikacij, tehnološka struktura, sodelovanje akterjev, izdelek, realizacija storitve, koordinacija poslovnega procesa, uporaba aplikacije, sodelovanje aplikacij, implementacija in namestitev. Z izborom zornega kota implicitno dobimo množico konceptov in relacij. Nadalje so aktivnosti še naslednje: izdelava in predstavitev modela, uporaba modela oziroma njegove predstavitve za komunikacijo z deležniki (*stakeholders*) in na koncu še vzdrževanje modela [36].

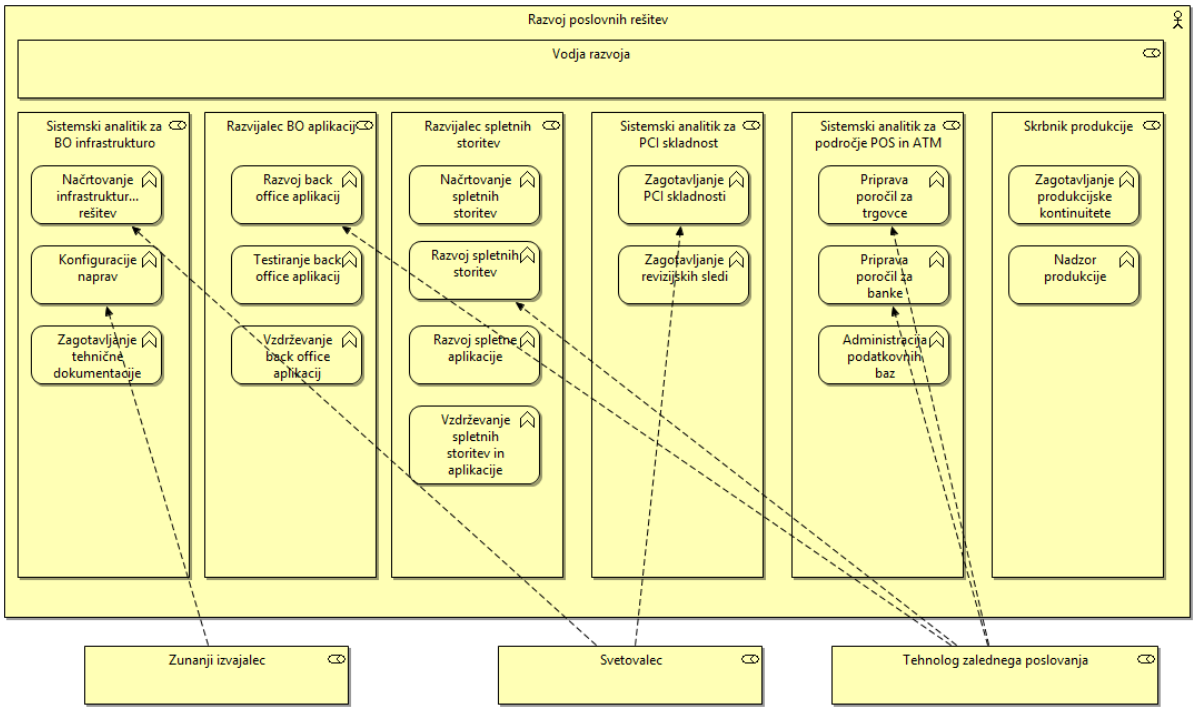
3.4.3 Primeri modelov ArchiMate

Naj za konec predstavitve jezika v tem razdelku podajam še dva primera modelov ArchiMate, ki predstavljata strukturo in zasnovo podjetja, obravnavanega v tem magistrskem delu.



Slika 5 - ArchiMate model visokonivojskih poslovnih funkcij

Na sliki 5 je prikazan model visokonivojskih poslovnih procesov, kot sem jih predstavil v razdelku 2.5.2., s pogledom iz zornega kota poslovnih funkcij in prikazom zunanjih vlog – na eni strani dobavitelji, zunanji izvajalci in poslovni partnerji, na drugi pa stranke: banke, trgovci in komitenti bank. Zunanji izvajalci svetujejo pri upravljanju družbe in podpornih procesih, poslovni partnerji sodelujejo pri podpori procesiranju, med tem ko dobavitelji izvajajo dobavo za podporo procesiranju in podporne procese. Podjetje prek poslovne funkcije procesiranja nudi storitve bankam, prav tako trgovcem in komitentom.



Slika 6 - Poslovna funkcija razvoja poslovnih rešitev

Funkcija razvoja poslovnih rešitev je prikazana na sliki 6 skozi prizmo organiziranosti poslovanja omenjene funkcije. V njem so identificirane vloge in za vsako od vlog podrobne poslovne funkcije, katere nosilci so omenjene vloge. Tako sistemski analitik za zaledno infrastrukturo načrtuje infrastrukturne rešitve, konfigurira naprave in zagotavlja tehnično dokumentacijo. Razvijalec zalednih aplikacij skrbi za razvoj, testiranje in vzdrževanje zalednih aplikacij. Razvijalec spletnih storitev načrtuje, razvija, in vzdržuje spletne storitve in aplikacije. Sistemski analitik za PCI skladnost le-to zagotavlja, zagotavlja pa tudi revizijske sledi.

3.4.4 ArchiMate kot orodje za modeliranje varnosti in upravljanja tveganj

V strokovnih in znanstvenih člankih je moč zaslediti predloge uporabe ArchiMate kot orodja za upravljanje tveganj. Tako Cholez in Feltus [37] ugotavljata, da so oddelki znotraj združbe močno povezani med seboj ter morajo biti v stalni interakciji za učinkovito delo in v primeru nekega dogodka (npr. izpada) v enem oddelku lahko pride do tveganja za izpad tudi v drugem. Tako je po njuno potreben sistematičen pristop k upravljanju tveganj, zato predlagata ogrodje za upravljanje tveganj informacijsko-komunikacijske tehnologije čez več oddelkov združbe, pri čemer uporabita arhitekturni model združbe v jeziku ArchiMate.

Jonkers [32] trdi, da je smiselno umestiti ERM v kontekst arhitekture podjetja (EA), ki poda celosten pogled na strukturo in zasnovo organizacije. Tako po njegovem ni presenetljivo, da EA vsebujejo poglavja in koncepte o tveganjih in varnosti, čeprav naj bi bilo še veliko prostora za izboljšave. Cholez in Feltus [37] ugotavljata podobno, ko pravita, da ArchiMate ni dovolj semantično bogat, da bi lahko modelirali vse elemente IT ekosistema.

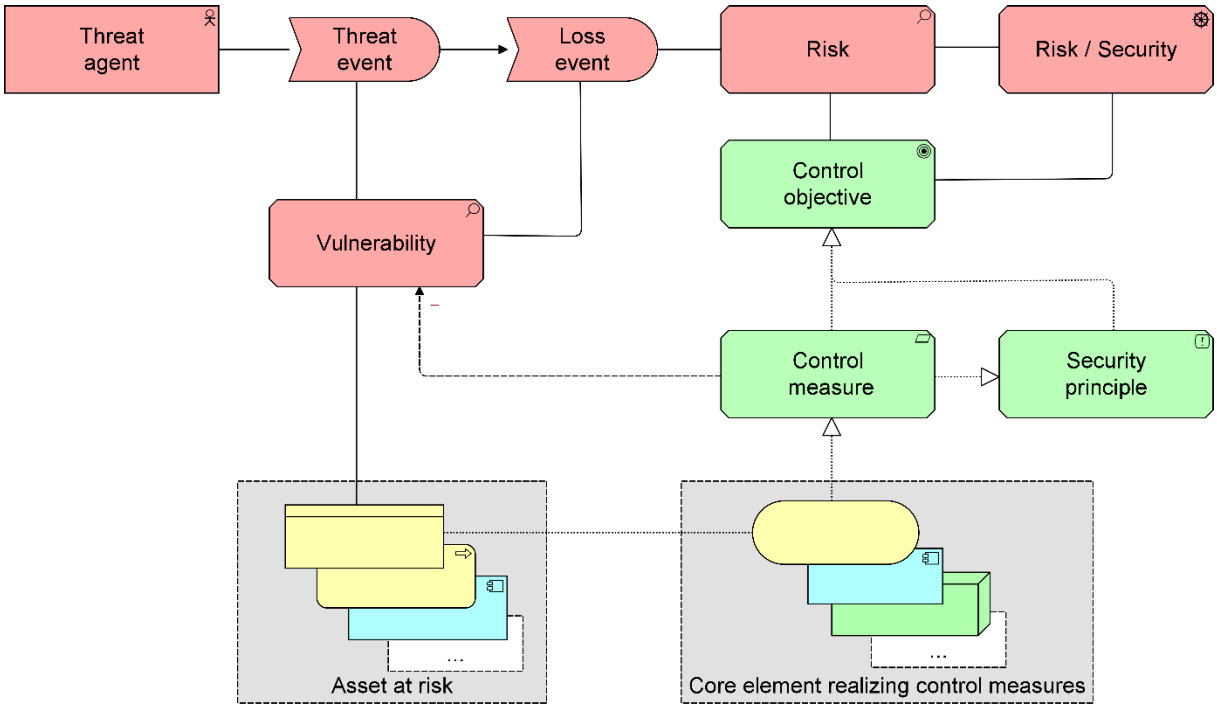
ArchiMate omogoča modeliranje varnostnih procesov in s tem omogoča podjetjem upravljati tveganja s ciljem [34]:

- oceniti tveganja in varovanja sredstev podjetja (tudi) zunaj informacijske varnosti,
- ponuditi navodila in tehnike za izdelavo varnostnih arhitekturnih modelov,
- ponuditi modelirne vzorce in načine izdelave za funkcije s področja varnosti, kot so preverjanje pristnosti (avtentikacija) in avtorizacija, revizijska sled, nadzor ipd.

Z namenom prikaza varnosti in tveganj na primerih modelov, ki sem jih naredil v tem magistrskem delu, bom v nadaljevanju opisal koncepte, ki jih opredeljuje jezik ArchiMate in opisal njihove lastnosti [4]. Spodaj podani koncepti so največkrat uporabljeni izrazi za tveganja in varnosti v najbolj uporabljanih varnostnih standardih in ogrodjih:

- Tveganje (*Risk*) – verjetnost, da se bo zgodila določena škoda, pri čemer gre za možnost, da bo izid dogodka drugačen od predvidenega
- Škodni dogodek (*Loss event*) - vsaka okoliščina, ki povzroči izgubo ali škodo določenega sredstva podjetja
- Grožnja (*Threat*) - nevarnost, da bi se izkoristila ranljivost in s tem povzročila morebitno škodo. ArchiMate natančneje loči dva koncepta:
    - Grozeč agent (*Threat agent*) – določen subjekt ali objekt, ki je zmožen povzročiti škodo bodisi namenoma (npr. heker pri vdoru v IS) bodisi ne namenoma (npr. napačno napisana programska koda, ki naredi škodo ali napačno zbrisani podatki iz baze)
    - Grozeč dogodek (*Threat event*) - dogodek, ki bi lahko vplival na premoženje podjetja
- Ranljivost (*Vulnerability*) – neka pomanjkljivost v sistemu, ki omogoča napadalcu ogroziti določeno sredstvo
- Domena (*Domain*) – množica sorodnih in/ali povezanih entitet, ki opredeljujejo semantiko iz določenega področja
- Domena tveganja (*Risk Domain*) – množica sorodnih in/ali povezanih entitet, ki si delijo podobne lastnosti v zvezi z varnostjo
- Ublažitev (*Risk Control, Treatment, Mitigation*) – množica varnostnih storitev za obrambo pred varnostnimi grožnjami
- Zahteva za nadzor (*Control Requirement*) – formalizirana potreba za nadzor za soočanje z identificirano grožnjo.
- Izpostavljeno sredstvo (*Asset at Risk*) – kakršnikoli podatki, naprave in druge premoženje podjetja, ki imajo povezavo z informacijami podjetja
- Politika (*Policy*) – množica pravil, ki opredeljujejo obnašanje sistema, ki obstajajo na različnih nivojih: strategija, upravljanje, načrtovanje in so različnega tipa: operativna, strukturna in vedenjska (obnašanje)

Vsi zgoraj opisani koncepti so prikazani na varnostnem meta modelu na sliki 7 spodaj.



Slika 7 - Uporaba varnostnih konceptov v jeziku ArchiMate
Vir: [32]

Uradna literatura [34] predlaga preslikavo konceptov tveganj in varnosti, opredeljenih zgoraj, s komponentami in koncepti jezika ArchiMate, kot je to prikazano v tabeli 1 spodaj.

Varnostni koncept (slo.)	Varnostni koncept (angl.)	Komponenta jezika ArchiMate
Škodni dogodek	Loss event	Business event
Grožnja	Threat	
Agent grožnje	Threat agent	več konceptov
Grozeč dogodek	Threat event	Business event
Tveganje	Risk	Assessment
Metrike tveganja	Risk metrics	Assessment attribute
Ranljivost	Vulnerability	
Domena	Domain	/
Domena tveganja	Risk domain	
Ublažitev	Risk Control, Treatment, Mitigation	več konceptov
Zahteva za nadzor	Control requirement	
Izpostavljeno sredstvo	Asset at risk	več konceptov
Politika	Policy	Principle
Cilj nadzora	Control objective	
Nadzorni ukrep	Control measure	

Tabela 1 - Pregled preslikave varnostnih konceptov s koncepti ArchiMate v barvah, kot sem jih uporabil v varnostnih modelih

3.4.5 Orodje Archi

Za modeliranje v ArchiMate so na voljo številna orodja, tako plačljiva kot brezplačna. Sam sem izbral programsko rešitev Archi, ki je odprtokodna in brezplačna programska oprema, izdana pod licenco

MIT[7], razvita v javanskem okolju Eclipse. Podprta je tudi s strani postavljalca standarda, The Open Group. Trenutno aplikacija Archi podpira ArchiMate jezik različice 2.1. Archi ponuja preprost in intuitiven uporabniški vmesnik, ki temelji na grafičnem uporabniškem vmesniku omenjenega okolja Eclipse.

3.4.6 ArchiMate modeli s področja varnosti in tveganj

V nadaljevanju poglavja sta izdelana dva ArchiMate modela s področja tveganj in varnosti, na podlagi katerih bom v nadaljevanju predstavil varnostne standarde in primere iz prakse pri poslovnih in IT procesih v podjetju. V naslednjih dveh razdelkih bom torej podal dva modela:

1. IT proces razvoja spletne aplikacije, pri čemer so identificirane varnostne pomanjkljivosti napake pri kodiranju, (ne)testiranje, manko preverjanja kode (*code-review*).
2. Poslovni proces uporabe spletne aplikacije za pridobivanje podatkov o komitentih s primarno poslovno funkcijo blokiranja kartic na zahtevo komitentov bank, pri čemer so identificirane varnostne pomanjkljivosti naslednje: ranljivost spletne aplikacije, škodljiva uporaba aplikacije s strani končnih uporabnikov – prikazovanje več PAN številk, ter uporaba aplikacije za ne-poslovne oziroma škodljive namene.

Varnostne koncepte sem v modelih obarval kot je to prikazano na sliki 7, torej koncepte grožnje (agent grožnje, grozeč dogodek), dogodek izgube, tveganje in ranljivost sem obarval rdeče, koncepte cilj nadzora, nadzorni ukrep in varnostni princip pa z zeleno barvo.

3.4.6.1 *IT proces razvoja spletne aplikacije*

V nadaljevanju je narisan in opisan model tveganj pri procesu razvoja programske opreme, modeliran le na poslovnem nivoju.

V oddelku razvoja poslovnih rešitev poteka razvoj in implementacija tako rešitev zalednih aplikacij, ki jih uporabljajo zaposleni kot tudi rešitve, ki ji podjetje ponuja bankam kot plačljive storitve in z njimi nastopa na trgu. Primeri aplikacij iz prve skupine so npr. program za konfiguracijo POS terminalov, program za pregled bankomatskih in POS transakcij ali npr. aplikacija za delo s karticami v klicnem centru. V drugi skupini rešitev, ki se jih ponuja (bančnemu) trgu, pa so med drugim rešitve za pripravo poročil o bankomatskem in POS prometu bankam in trgovcem, spletni portal za trgovske izpiske, spletne storitve (*web services*) in spletna aplikacija za banke. Prav slednjo bom uporabil za oba modela ArchiMate v tem poglavju.

Značilno se vsak razvoj začne z zahtevkom za spremembo (zahtevek za spremembo je zahteva, ki določa, kaj naj bo na novo razvito, dopolnjeno ali popravljeno), ki ga navadno poda tehnologija – vsebinski skrbniki aplikacij, sistemov oziroma programskih rešitev. Gre namreč za to, da je večina zahtevkov za spremembo vsebinske narave; če pa je tehnične, lahko zahtevek poda tudi razvijalec oziroma sistemski analitik. Omenjeni zahtevki se podajo v sistem za upravljanje sprememb, implementiranem v spletni aplikaciji Bugzilla[8].

---

[7] MIT je ena od odprtokodnih licenc, ki je najbolj odprtega značaja. Dovoljeno je vsakršna uporaba programske opreme, izdane pod to licenco, brez zaračunavanja, dovoljeno je neomejeno kopiranje, predelava, modifikacije, objava, distribucija, spajanje, podlicenciranje. Vse, kar licenca dopušča, mora nujno vsebovati izjavo o avtorstvu in dovoljenje za uporabo.

[8] Bugzilla je odprtokodna spletna programska oprema, ki je v prvi vrsti namenjen sledenju reševanja problemov in napak v programski in strojni opremi (*issue-tracking system*). Napisana je v programskem jeziku Perl. Zgodovinsko gledano je Bugzilla, tako kot je njen osnovni namen, v podjetju služila kot orodje za sledenje in odpravo zahtevkov in napak v informacijski infrastrukturi in aplikacijah. Kasneje, skozi faze vpeljave izboljšav in novih postopkov, ki jih določata ogrodje ITIL in standard ISO/IEC 20000, je bila v Bugzillo dodana cela vrsta prilagoditev. Tako je v njej možno dodajanje dokumentov, vnašanje ključnih besed, pisanje datumov rokov določenih aktivnosti, potrjevanje aktivnosti z zastavicami in drugo potrebno za podprtje dela po prej omenjenih standardih.

Zahtevek za spremembo mora biti najprej preverjen in nato odobren in potrjen s strani vodje oddelka. V zahtevku so zabeleženi podatki, kot so odgovorni nosilec in odgovorni izvajalec naloge, rok izvedbe zahtevka, sistem in komponenta, kateremu pripada zahtevek, trenutni status aktivnosti, dokumentacija o zahtevku, specifikacija in ostali pomembni podatki o posameznem zahtevku.

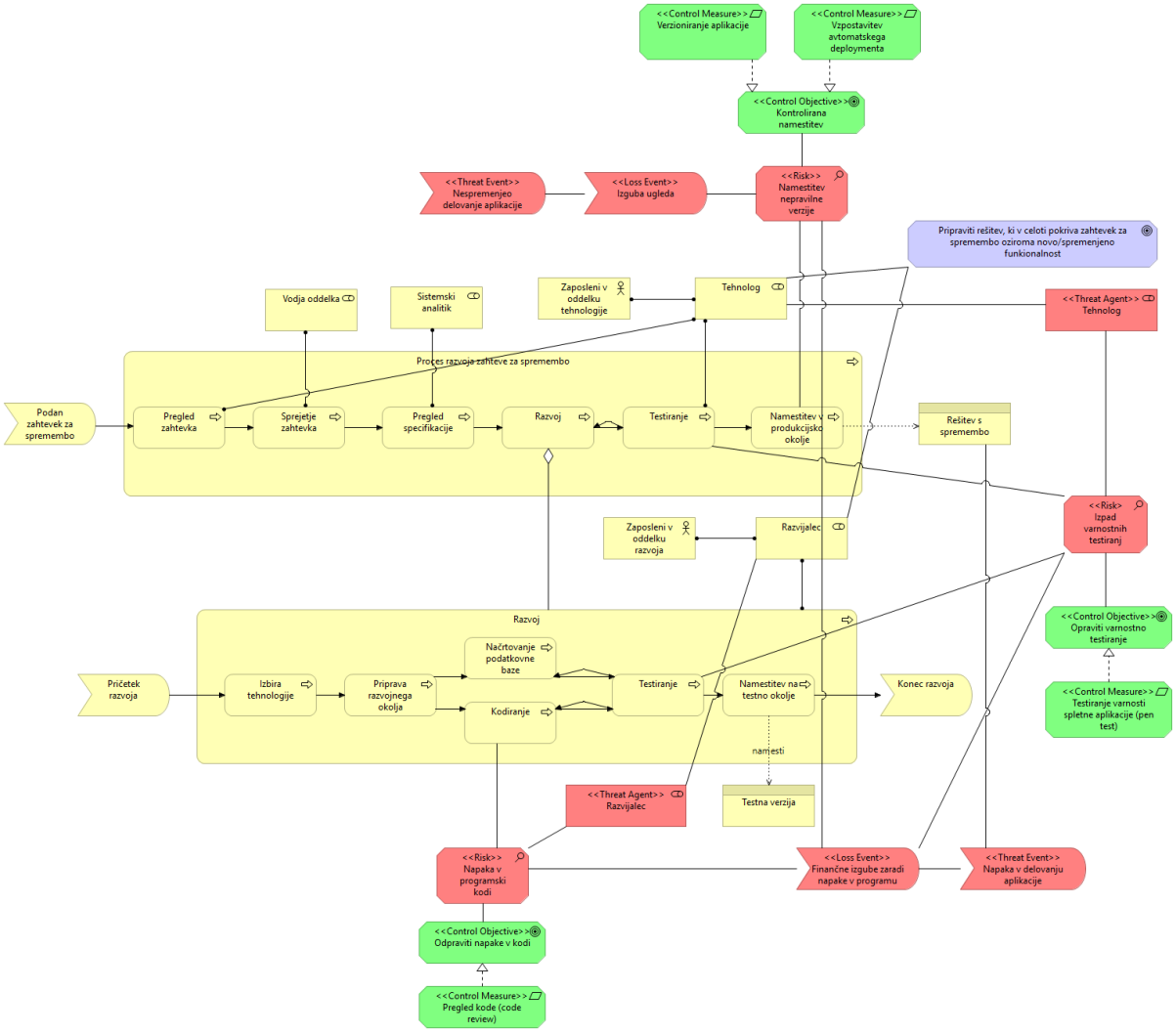
Sledi pregled specifikacije in morebitni zaključni popravki, preden se sam razvoj lahko začne.

Razvoj je proces, ki v povprečju traja največ časa in zahteva največ virov. Začne se po potrjeni in dokončno sprejeti specifikaciji. V prvi fazi je pomembno izbrati pravo tehnologijo (če je to potrebno v primeru večje spremembe ali nove zahteve; v primeru dodatne funkcionalnosti ali spremembe/popravka obstoječe izbira tehnologije ni potrebna). Z izbiro tehnologije se odpre vprašanje izbire razvojnega okolja, v katerem poteka v nadaljevanju proces postavitve podatkovnega modela in samo kodiranje oziroma implementacija funkcionalnosti. Ko je proces kodiranja zaključen, sledi proces testiranja in sicer IT testiranja, to je testiranja, ki se ga opravi avtomatsko z *unit*[9] ali *CUIT*[10] testiranjem ali pa ga opravi razvijalec. Testiranje je aktivnost, ki lahko v primeru nepravilnih ali neustreznih rezultatov vrne proces nazaj v fazo kodiranja in razvoja podatkovne baze. Ko je razvojno testiranje zaključeno, se izvede namestitev rešitve na testno okolje, kjer se izvaja vsebinsko testiranje, ki ga opravi tehnolog. V primeru neustreznih rezultatov se lahko ponovno izvede proces razvoja. Ob potrjenem testiranju s strani tehnologije se lahko s produkcijsko spremembo izvede namestitev na produkcijsko okolje. S tem je proces razvoja zaključen in izdelek oziroma *output* je zahtevana sprememba (tj. dopolnitev funkcionalnosti, nova funkcionalnost, popravek napake ali nepravilnega delovanja) v produkcijskem okolju.

Tveganja, ki jih lahko med drugimi identificiramo pri procesu razvoja spletne aplikacije, so lahko naslednja:

T1.1. Napaka v programski kodi
T1.2. Manko varnostnega testiranja
T1.3. Namestitev nepravilne verzije aplikacije

[9] UNIT testiranje je avtomatsko ali ročno testiranje, pri katerem se testira manjše, neodvisne enote (*unit*) programske kode. Navadno se uporablja pri pristopu ekstremnega programiranja.
[10] CUIT ali *coded user interface testing* je način testiranja programske opreme, ki temelji na preverjanju rezultatov preko uporabniškega vmesnika.



Slika 8 - Model tveganj pri IT procesu razvoja programske opreme (povečana slika je na voljo v dodatku 10.2.1)

T1.1 Napaka v programski kodi

Napake v programski kodi (tudi hrošči, angl. *bugs*) so sestavni del procesa razvoja programske opreme. So lahko posledica več različnih razlogov, kot so pomanjkljive/napačne specifikacije, napake razvijalca, malomarnost, napačne uporabe struktur, napake v samem ogrodju okolja ali *API*-ju[11] idr. Obseg napak se lahko zmanjša s tem, da kodo napiše izkušenejši razvijalec, da kodo že med samim razvojem sproti preverimo, uporabimo *unit* teste in seveda celovito testiramo. V vsakem primeru pa, kljub takšni težnji, 100% pravilnega delovanja brez programskih napak ne moremo zagotoviti.

Iz slike 8 je videti, da sem kot cilj nadzora tveganja napak v programski kodi identificiral zmanjšanje napak v kodi z ukrepom izvajanja postopka pregleda kode (*code review*). Pregled kode je proces, pri katerem je bistvo v tem, da izvorno kodo pregleda nekdo ali nekaj, ki ni avtor te kode. S tem se lahko prepoznajo napake, ki jih sicer izvorni avtor ni zaznal, pri čemer lahko pregled poteka že med samim razvojem (*on-the-fly*), kjer dvojica razvijalcev programira za isto delovno postajo in se ukvarja z isto kodo (to poznano iz pristopa agilnih programerskih tehnik, kot je ekstremno programiranje) ali pa je pregled kode izveden naknadno. Pri tem načinu drugi razvijalec (ali več njih) dobi kodo, kot je ta na

[11] API (*Application Programming Interface*) je programska oprema, ki omogoča interakcijo z drugo programsko opremo. Največkrat ponuja množico metod/funkcij in razredov, s katero lahko uporabljamo omenjeno programsko opremo oziroma komponento [10].

pisana, in odvisno od intenzitete preglede opravi manj ali bolj podroben pregled. Pri tem lahko dobi seznam sprememb in preveri, ali je koda v skladu z omenjenim seznamom, lahko opravi pregled sloga programiranja, preveri uporabljene strukture in metode, opravi varnostno analizo in pregleda samo sintakso. Za ta namen so na voljo namenska orodja, najmanj kar je potrebno, pa je program za primerjavo prejšnje verzije kode s spremenjeno.

Standard in poglavje, ki zajema ublažitev opisane ranljivosti:

- ITIL – ISO/IEC 20000: Change management
- PCI DSS: Requirement 6: Develop and Maintain Secure Systems and Applications (podrobneje v poglavju 6.3.2)

T1.2 Manko varnostnega (penetracijskega) testiranja

Testiranje aplikacij je obsežen in izredno pomemben del procesa razvoja programske opreme. Obstaja vrsta modelov in paradigem testiranja, ogrodij in metodologij ter mnogo orodij, tako prosto dostopnih kot tudi plačljivih, ki so na voljo za izvajanje različnih vrst testiranja.

Ker je, kot že rečno, testiranje obsežno področje, se v nadaljevanju (pričakovano) osredotočam le na tveganje neizvajanja varnostnega (penetracijskega) testiranja spletne aplikacije, ki ga ISO opredeljuje kot vrsto testiranja, ki izvede oceno stopnje zaščite testirane entitete (objekta) s povezanimi podatki in informacijami pred tem, da jih neavtorizirane osebe in sistemi ne morejo uporabiti, brati ali spreminjati in da neavtorizirane osebe in sistemi nimajo dostopa do teh podatkov in informacij.

Varnostno testiranje v obliki penetracijskih (*pen*) testov je ena najstarejših metod za oceno varnosti računalniških sistemov, saj je Ministrstvo za obrambo ZDA že v zgodnjih 70 letih 20. stoletja uporabilo to metodo za prikaz varnostnih pomanjkljivosti v računalniških sistemih in začelo s programom razvoja bolj varnih sistemov. *Pen* testiranje je namenjeno temu, da združbe zagotovijo varnost svojih informacijskih sistemov s katerimi se varnostne pomanjkljivosti lahko odpravijo še preden bi morda bi lahko bile izpostavljene. Cilj penetracijskega testiranja je identifikacija in poročilo o varnostnih ranljivostih, s čimer omogočimo združbi odpraviti težave in pomanjkljivosti na načrtovan način in s tem bistveno povečati nivo zaščite združbe [38]. Varnostno testiranje spletne aplikacije v obliki *pen* testov lahko opravlja zunanji, neodvisni izvajalec.

Pomembno je ločiti med penetracijskim testiranjem in skeniranjem ranljivosti. Namen penetracijskega testiranja je namreč simulacija napada iz resničnega sveta s ciljem, kako daleč lahko napadalec prodre v okolje, ki se testira [39]. S tem se omogoči boljše razumevanje morebitne izpostavljenosti in pripravi strategijo za obrambo pred napadi. Penetracijski test je aktiven proces, ki lahko vključuje izkoriščanje ugotovljenih pomanjkljivosti. Na drugi strani je skeniranje ranljivosti avtomatski proces in je lahko eden od prvih korakov penetracijskega testiranja, saj lahko rezultate tega skeniranja izvajalec testiranja (tester) uporabi za načrtovanje strategije testiranja. Tudi če skeniranje ranljivost ne zazna znane ranljivosti, bo izvajalec testiranja pogosto pridobili dovolj znanja o sistemu za identifikacijo morebitnih vrzeli varnosti. Penetracijsko testiranje je predvsem ročni proces, pri katerem tester uporablja predvsem svoje znanje sistema s katerim poskuša vdreti v okolje. Se pa lahko pri tej dejavnosti uporabljajo tudi avtomatska orodja.

Združba se mora zavedati, da je internet vseskozi spreminjajoče se področje, kakor to velja tudi za varnost na internetu. S tega vidika uspešno penetracijsko testiranje še ne zagotavlja, da je združba varna pred vsako obliko napada. Kot trdi Alisherov [38], 100 odstotnega varnostnega testiranja ni mogoče doseči. Tako ni mogoče testirati ranljivosti v programski opremi ali sistemih, ki še niso znana v času testiranja ali testirati matematično celotnega nabora vseh možnih vhodov oziroma izhodov za vsako komponento programske opreme, ki se uporablja.

Na sliki 8 je torej prepoznano tveganje izpada varnostnega testiranja, ki je vezan na poslovni proces testiranja programske opreme tako pri procesu razvoja zahteve za spremembo kot tudi znotraj podprocesa samega razvoja. Cilj nadzora tveganja manka varnostnega testiranja je seveda opraviti varnostno testiranje. Vezano na to tveganje je ukrep nadzora izvesti testiranje varnosti spletne aplikacije z načinom penetracijskega testiranja (*pen* test).

Standard in poglavje, ki zajema ublažitev opisane ranljivosti:

- PCI DSS: Requirement 11: Regularly test security systems and processes (podrobneje v poglavju 11.3)

T1.3 Namestitev nepravilne verzije aplikacije

Tretje prepoznano tveganje je namestitev nepravilne verzije aplikacije v produkcijsko okolje, kar je lahko posledica ročnega nameščanja verzij. Nameščanje programskih rešitev v produkcijsko okolje (*deployment*) je pomemben proces, sestavljen iz več aktivnosti. V našem primeru je nameščanje sestavljeno iz kombinacije (ali vseh) naslednjih korakov: arhiviranje obstoječe verzije in arhiviranje podatkovne baze za morebitni primer ponovne povrnitve v prejšnje stanje, nameščanje nove verzije knjižnic, nameščanje nove verzije programa, posodobitev konfiguracij, posodobitev podatkovne baze, kratek test delovanja.

Pri procesu nameščanja, še posebej če gre za ročno nameščanje, je možnih veliko napak, kot so napaka pri prenosu podatkovnega modela ali podatkov iz podatkovne baze iz testnega/simulacijskega okolja v produkcijo, napaka pri konfiguraciji in parametrizaciji aplikacije, napačno kopiranje binarnih datotek na namestitvene mape ipd. Vse te napake so lahko posledica nedokumentiranosti postopkov, pomanjkljivega ali sploh manjka plana implementacije namestitve ter človeških napak, pa tudi neznanja, neprevidnosti in malomarnosti.

Poleg samega opisanega procesa namestitve je pomembno tudi okolje in podporne aktivnosti, kot so pravilno verzioniranje oziroma označevanje posameznih izdaj izdanih aplikacij, formalizirani in dokumentirani postopki namestitev, pravice in vloge pri nameščanju, skratka vse za kontrolirane in dokumentirane aktivnosti nameščanja.

Napake pri nameščanju lahko zmanjšamo s kontroliranimi in dokumentiranimi postopki; kot primera sta v diagramu ArchiMate (Slika 8) navedena uvedba verzioniranja in vzpostavitev avtomatskega nameščanja z ustreznimi orodji, ki so na voljo na tržišču.

Posledica napak pri nameščanju so lahko manjšega obsega in vpliva na delovanje storitev, lahko pa pride do večjih izpadov storitev in posledično večje škode, ki se lahko kaže tudi v finančnih posledicah za združbo.

Standard in poglavje, ki zajema ublažitev opisane ranljivosti:

- ITIL – ISO/IEC 20000: Change management

3.4.6.2 *Poslovni proces uporabe spletne aplikacije*

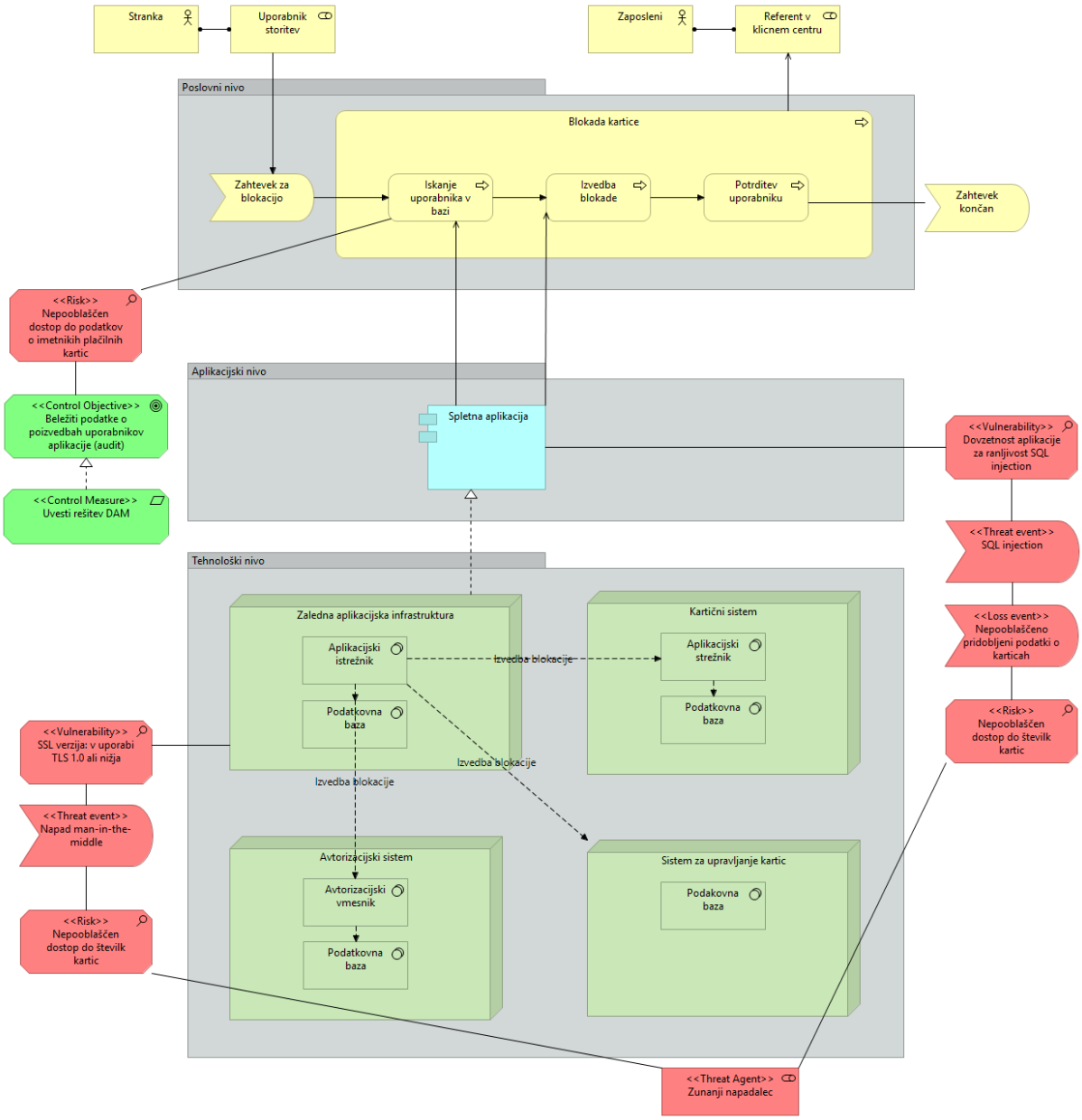
Poslovni proces uporabe spletne aplikacije za pridobivanje podatkov o komitentih s primarno poslovno funkcijo blokiranja kartic na zahtevo komitentov bank je modeliran v diagramu na sliki 9. Za razliko od modela, predstavljenega v prejšnjem razdelku, sega ta prikaz varnostnih tveganj čez vse tri nivoje jezika ArchiMate, tako da imamo tu uskladitev poslovnega, aplikacijskega in tehnološkega nivoja.

Spletna aplikacija ima poleg ostalih modulov, kot so pregled avtorizacijskih logov ATM in POS prometa, pregled poročil za banke, dostop do dokumentacije za banke in trgovce, tudi modul za delo s podatki o imetnikih kartic. V tem modulu je možno preveriti osnovne in razširjene podatke o karticah ter zgodovino podatkov, spreminjati limite, spreminjati status in blokirati kartice. Na diagramu je

modeliran poslovni proces preklica oziroma blokade kartice na zahtevo komitenta banke. Komitent v primeru kraje, izgube ali drugega razloga lahko pokliče v klicni center, kjer referent sprejme njegov klic in na podlagi podanega razloga in preverjanja osebnih podatkov izvede blokado. V spletni aplikaciji lahko po različnih parametrih poišče kartico, ki je predmet blokade. Ti parametri so lahko sama številka kartice, ime in priimek, datum rojstva, davčna številka, številka računa ali enotna matična številka občana. Ko kartico najde, se na zaslonu za spremembo statusa kartice pojavi padajoči meni, kjer so na voljo različni statusi kartice: aktivna kartica, splošna blokada, poškodovana kartica, izgubljena kartica, blokada računa ipd. Z izborom blokade in potrditvijo se zahteva pošlje v zaledni sistem, kjer se izvede dejanska sprememba statusa. Ob uspešni spremembi se referentu v klicnem centru pojavi obvestilo in s tem sporoči komitentu, da je blokada uspešno končana.

Z aplikacijskega vidika se zgornji proces, kot sem že omenil, izvede v spletni aplikaciji, ki je uporabniku na voljo v spletnem brskalniku na njegovi delovni postaji.

Zahtevek za poizvedbo o podatkih imetnikov plačilnih kartic in sama sprememba statusa kartice gre iz spletne aplikacije v zaledni sistem. Zaledni sistem je iz vidika procesiranja blokade tehnološko sestavljen iz več podsistemov. Zahtevek namreč prevzame sistem spletnih storitev, ki glede na podane parametre kartice (kot so tip kartice (debetna, kreditna), produkt kartice (Maestro, Visa, American Express, Karanta,…), lastnik kartice (banka lastnica) ipd.) pošlje zahtevke za spremembo statusa v različne podsisteme. Blokada se lahko zabeleži v avtorizacijski sistem, ki je glavni računalniški sistem za procesiranje bankomatov in pos terminalov. Lahko se zabeleži tudi v podatkovni bazi kreditnih kartic. Nenazadnje se vedno zabeleži tudi v interni aplikaciji, ki je glavna avtoriteta za podatke o karticah in njihovih imetnikih. Vsak od omenjenih podsistemov služi svojemu namenu in je implementiran na različen način: tehnološko, tehnično in popolnoma neodvisno od drugega.



Slika 9 - Model tveganj pri poslovnem procesu uporabe spletne aplikacije (povečana slika je na voljo v dodatku 10.2.2)

Tveganja, ki jih lahko med drugimi identificiramo pri procesu uporabe spletne aplikacije, so lahko naslednja:

T2.1. Nepooblaščeni vpogledi v podatke o imetnikih plačilnih kartic (grožnja z notranje strani)
T2.2. Poskus napada z SQL vrivanjem (*SQL injection*)

T2.1 Nepooblaščeni vpogledi v podatke o imetnikih plačilnih kartic

Zaposleni v združbi, ki delajo na področju operative v klicnem centru ali npr. na oddelku vnosa, tako imenovani referenti, imajo pristojnost, da na zahtevo lahko izvedejo vpogled o finančnem stanju določene osebe, pregledajo avtorizacije in transakcije te osebe, izvedejo spremembe statusov kartic in spremembe limitov. Vse to so seveda podatki zaupne ali strogo zaupne narave. Uporabnik lahko tako pridobi podatke za osebe, ki niso predmet poslovnih zahtev ali potreb in tako lahko zlorabi podatke in informacije, ki jih je pridobil.

Način, s katerim lahko preprečimo ali vsaj zmanjšamo možnost zlorab, je dostopnost do podatkov na kar se da majhen nabor. Do toliko podatkov, kolikor za nemoteno delo zaposleni (v našem primeru referent) potrebuje. Drugi način, ki neposredno ne prepreči zlorabe, pač pa v primeru suma lahko potrdi ali ovrže nepooblaščene dostope do zaupnih podatkov, poskrbi za sledljivost dejanj in tudi dostopov do podatkovne baze, je revizijska sled v podatkovni bazi. Revizijsko sled zagotavljajo DAM sistemi.

Standard in poglavje, ki zajema ublažitev opisane ranljivosti:

- PCI DSS: Requirement 7: Restrict Access to Cardholder Data by Business Need-to-Know (podrobneje v poglavju 7.1 in 7.2)
- PCI DSS: Requirement 10: Track and Monitor All Access to Network Resources and Cardholder (podrobneje v poglavju 10.1, 10.2, 10.3, 10.5, 10.7)

T2.2 Poskus napada z SQL vrivanjem (*SQL injection*)

Spletna aplikacija, ki nudi finančne podatke strankam, je lahko zaradi svoje narave podvržena zunanjim napadom. Razširjenih je veliko vrst napadov na tovrstne aplikacije. Vrste napadov dobro klasificira preglednica OWASP TOP 10, seznam najnevarnejših spletnih ranljivosti, ki jo pripravlja združba OWASP[12]. Preglednica nastaja na vsakih nekaj let, posamezna vrsta ranljivosti pa je razvrščena glede na aktualno zaznano pogostnost. Trenutna aktualna verzija je OWASP TOP 10 – 2013. Na njej najdemo ranljivosti kot so vrivanje, podtikanje skript, napaka pri overjanju in upravljanju sej, nezavarovan neposreden dostop do objektov, potvarjanje spletnih zahtevkov, napake v varnostnih nastavitvah, nezadostna zaščita hrambe kriptografskih podatkov, neprimerna zaščita neposrednega dostopa do povezave, nezadostna zaščita podatkov pri prenosu, nepreverjanje preusmeritve brskalnika ipd.

SQL vrivanje je ena od podzvrsti vrivanja, ki je v aktualni preglednici uvrščena na prvo mesto. Značilno zanj je, da napadalec poizkuša izvesti škodljiv SQL stavek preko vhodnih podatkov iz aplikacije odjemalca, največkrat iz spletne aplikacije. S takim škodljivim SQL stavkom lahko napadalec pridobi občutljive ali zaupne podatke, lahko spremeni vrednosti v bazi, lahko spremeni strukturo baze ipd. SQL vrivanje se zgodi, ko sta izpolnjena dva pogoja: podatki pridejo v program iz nepreverjenega vira in hkrati so ti podatki uporabljeni za dinamično konstruiranje SQL poizvedbe oziroma ukaza.

V našem konkretnem identificiranem varnostnem tveganju lahko napadalec pridobi podatke o plačilnih karticah, pri čemer izkoristi ranljivost aplikacije v obliki dovzetnosti aplikacije za SQL vrivanje.

Standard in poglavje, ki zajema ublažitev opisane ranljivosti:

- PCI DSS: Requirement 6: Develop and Maintain Secure Systems and Applications (podrobneje v poglavju 6.3, 6.3.2, 6.5, 6.5.1, 6.5.2 – 6.5.10, 6.6)

3.4.6.3 *Prepoznana varnostna tveganja*

V tem poglavju so za pregled v tabeli 2 še enkrat zbrana vsa prepoznana varnostna tveganja pri izbranih IT in poslovnih procesih, ki sem jih prikazal na modelih ArchiMate v prejšnjih razdelkih tega poglavja. Tveganja so označena z identifikatorji (TX.X), na katere se sklicujem v nadaljevanju tega magistrskega dela. Vsako od tveganj ima ob sebi zabeležen še izbran standard oziroma ogrodje in številko poglavja oziroma proces, ki to ranljivosti odpravlja.

[12] OWASP je neprofitna, dobrodelna združba, ki se ukvarja in promovira razvoj varne programske opreme.

Tveganje	Ogrodje/standard in poglavje, ki odpravlja tveganje/ranljivost
IT proces razvoja spletne aplikacije	
T1.1. Napaka v programski kodi	ITIL/ISO/IEC 2000: Change management
	PCI DSS poglavje 6.3.2
T1.2. Manko varnostnega testiranja	PCI DSS poglavje 11.3 (11.3.1, 11.3.2, 11.3.3, 11.3.4)
T1.3. Namestitev nepravilne verzije aplikacije	ITIL/ISO/IEC 20000: Change management
Poslovni proces uporabe spletne aplikacije	
T2.1. Nepooblaščeni vpogledi v podatke o imetnikih plačilnih kartic (grožnja z notranje strani)	PCI DSS poglavje 2.2
	PCI DSS poglavje 7.1 in 7.2
	PCI DSS poglavje 8.1 (8.1.4) in 8.5
	PCI DSS poglavje 10.1
	PCI DSS poglavje 10.2 in 10.3
	PCI DSS poglavje 10.5
	PCI DSS poglavje 10.7
T2.2. Poskus napada z SQL vrivanjem (*SQL injection*)	PCI DSS poglavje 6.3
	PCI DSS poglavje 6.3.2
	PCI DSS poglavje 6.5
	PCI DSS poglavje 6.5.1
	PCI DSS poglavje 6.5.2 – 6.5.10
	PCI DSS poglavje 6.6

Tabela 2 - Prepoznana varnostna tveganja pri izbranih IT in poslovnih procesih

4 Standard PCI DSS

4.1 Predstavitev standarda PCI DSS

Potreba po varovanju podatkov o imetnikih plačilnih kartic je največje izdajatelje plačilnih kartic, to so Visa Inc., Mastercard Inc., American Express Co., Japan Credit Bureau (JCB) in Discover Financial Services, privedla k ustanovitvi sveta Payment Card Industry Security Standards Council, ki je opredelil Payment Card Industry Data Security Standard (kratica PCI DSS) [1]. Ta vodi podjetja, ki ponujajo storitve, povezane s plačilnimi karticami in trgovce, da implementirajo boljšo varnostno infrastrukturo in ureja področje rokovanja s podatki o plačilnih karticah ter s tem zmanjšajo možnost za varnostne kršitve. Lahko rečemo, da vsebuje nabor smernic oziroma zahtev, tako tehničnih kot operativnih, za zagotavljanje varnega ravnanja z zaupnimi podatki. Prva verzija je nastala leta 2004, trenutna verzija pa je 3.1, izdana aprila leta 2015.

Standard je razdeljen v šest sklopov, ki vsebuje dvanajst krovnih zahtev oziroma poglavij, ki vključujejo upravljanje varnosti, politike, postopke, mrežno arhitekturo, načrte programske opreme in ostale ukrepe. Spodaj je podan pregled standarda na najvišjem nivoju, ki zajema omenjenih 12 krovnih poglavij [39], nekatera so zelo tehnično naravnana, nekaj pa je bolj procesno orientiranih:

- Izdelava in vzdrževanje varnega omrežja
  1. Protipožarne pregrade morajo biti nameščene in redno vzdrževane.
  2. Prepovedana je uporaba privzetih sistemskih gesel in drugih varnostnih parametrov.
- Ščitenje podatkov o imetnikih plačilnih kartic
  3. Ščititi je potrebno shranjene podatke o imetnikih plačilnih kartic.
  4. Potrebna je enkripcija pri prenosih podatkov o imetnikih plačilnih kartic preko javnih, odprtih omrežij.
- Vzdrževanje programa za upravljanje ranljivosti
  5. Uporabljati in redno nadgrajevati je potrebno protivirusne programe.
  6. Razvijati in vzdrževati je potrebno varne sisteme in aplikacije.
- Izvajanje strogih ukrepov za nadzor dostopa
  7. Omejiti je potrebno dostop do podatkov o plačilnih sredstvih na minimum, ki še zagotavlja nemoteno delovanje združbe.
  8. Vsaka oseba naj ima edinstveno identifikacijo pri dostopih do sistemov in aplikacij.
  9. Omejiti je potrebno fizični dostop do podatkov o imetnikih plačilnih kartic.
- Redno nadzorovanje in testiranje omrežij
  10. Slediti in nadzorovati je potrebno vse dostope do mrežnih virov in podatkov o imetnikih plačilnih kartic.
  11. Redno je potrebno testirati varnostne sisteme in procese.
- Vzdrževanje politike informacijske varnosti
  12. Potrebno je vzdrževati politiko, ki vključuje informacijsko varnost.

PCI DSS skladne morajo biti vse entitete, ki ne glede na njihovo velikost, procesirajo, shranjujejo, prenašajo ali kakorkoli operirajo s podatki o kreditnih karticah in njihovih lastnikih. Ti podatki so naslednji: številka kartice (*primary account number - PAN*), ime lastnika kartice (*cardholder name*), datum veljavnosti (*expiration date*), storitvena koda (*service code*). Poleg naštetih morajo ustrezno operirati še z občutljivimi avtentikacijskimi podatki, kot so: sledljivostni podatki (*full track data – magnetni zapis ali čip na kartici*) in CAV2/CVC2/CVV2/CID[13] podatki. Omenjene entitete, ki morajo biti PCI DSS skladni, so lahko trgovci, procesorji plačilnega prometa, banke, ponudniki storitev ipd.

[13] Podatki varnostne narave. Card Validation Value (CVV) je posebna vrednost, enkodirana na magnetnem zapisu na kartici, ki potrjuje, da je kartica fizično prisotna. CVC je Mastercardova različica CVV zapisu. Card Validation Value 2 (CVV2) je zapis, natisnjen na kartici z namenom potrjevanja, da je kartica fizično prisotna

Glavni podatek, ki ga je torej v sklopu PCI DSS potrebno varovati, je PAN, številka kreditne kartice oziroma številka debetne kartice. To je tako občutljiv podatek, da jih moramo v podatkovnih bazah in tudi ostalih sistemih za shranjevanje podatkov hraniti v kriptirani obliki. To dosežemo s tokenizacijo, postopkom ščitenja PAN številk. Pri obdelavi v informacijskih sistemih sme nastopati samo kriptirana oblika PAN-a, torej njegov *token*[14]. Seveda je poleg tega treba varovati tudi ostale podatke, predvsem osebne podatke o imetnikih kartic, številke računov, stanja na njihovih računih in podobno.

Pridobitev certifikata, ki potrjuje, da je podjetje v skladu s standardom, zahteva izpolnitev vseh točk iz podrobnega seznama zahtev. Skladnost ocenjujejo kvalificirani varnostni ocenjevalci (*PCI Qualified Security Assessor - QSA*). Certifikat, ki se uradno imenuje *Certificate of PCI DSS Compliance*, se pridobi enkrat, potem pa ga je potrebno vsako leto obnavljati zaradi morebitnih sprememb v samem standardu ali sprememb v operativnem okolju združbe [40].

PCI skladnost za združbe zagotavlja, da se kljub morebitni realizaciji grožnje združbe izognejo maksimalnim kaznim. Če kakšno podjetje ne uspe s certifikacijo, so lahko oglobljene, lahko jih doletijo še finančne kazni, umik POS opreme, izguba članstva pri kartičnem poslovanju (s strani glavnih procesorjev kartic), negativna publiciteta, izgubljeno zaupanje strank in dobaviteljev, zmanjšanje prihodkov od prodaje ipd., v najslabšem primeru pa prenehajo z opravljanjem dejavnosti [40].

Za podjetje, kot pravi Nicho [41], PCI skladnost ne pomeni, da je izolirano od vseh možnih groženj in goljufij, kvečjemu integracija relevantne varnosti in vpeljave ogrodij ali standardov omogoča preprečitev tveganja večjih razsežnosti.

4.2 Primer implementacije PCI standarda za področje revizijske sledi

V razdelku 3.4.6.2 sem za uporabo spletne aplikacije za delo s podatki o imetnikih plačilnih kartic identificiral varnostno tveganje nepooblaščenega dostopa do (zaupnih) podatkov (tveganje T2.1). Kontrola, s katero (lahko) zmanjšamo omenjeno identificirano tveganje, je beleženje aktivnosti uporabnikov v aplikaciji. Iz tega razloga sem za namen prikaza implementacije PCI DSS v praksi izbral področje podatkovnih baz, še podrobneje pa je opisan izbor DAM rešitve, sistema, ki beleži aktivnosti in dostope končnih uporabnikov do podatkov v podatkovnih bazah.

4.2.1 PCI in podatkovne baze

V prejšnjem razdelku opisane točke zajemajo celoten nabor zahtev, v nadaljevanju se bom omejil le na del standarda, ki opisuje zahteve s področja podatkovnih baz, ki jih je Mike Chapple v svojem članku na spletni strani databases.about.com [42] povzel v nekaj bistvenih točkah in v nadaljevanju jih bom povzel tudi sam:

- Potrebno je imeti opredeljeno politiko za varovanje podatkov o imetnikih plačilnih kartic, jo razvijati in vzdrževati.
- Podatkovna baza se mora nahajati znotraj mrežnega segmenta podjetja, torej v internem okolju, stran od *dmz* (demilitariziranega) segmenta, ki je stik zunanjega sveta z omrežjem podjetja, ki hrani podatke. Vse poskuse dostopa do podatkovne baze z nezaupanja vrednih omrežji je potrebno onemogočiti. Obvezno je uporabljati zasebne IPje za strežnike s podatkovnimi bazami.
- Zamenjati je potrebno privzeta gesla, ki jih dostavi izdelovalec programskih rešitev in podatkovnih baz; zagotoviti je potrebno uporabo močnih gesel za vse uporabniške račune in kot že omenjeno, zamenjavo vseh privzetih gesel.

---

pri uporabniku. Card Validation Code 2 (CVC2) je Mastercardov ekvivalent CVV2, medtem ko je Card Identification Data (CID) ekvivalent American Expressa in Discoverja za zapis CVV2.

[14] Token v IT lahko opredelimo kot predstavitveni nadomestek podatka, ki ga ščitimo.

- Vse ne-konzolne administrativne dostope je potrebno enkriptirati s katerim od kakovostnih algoritmov za enkripcijo, kot so VPN, SSL, ssh in podobni. S tem zmanjšamo možnost vohljanja za administrativnimi pravicami za dostop do podatkovne baze.
- Hranjenje podatkov o imetnikih plačilnih kartic v podatkovnih bazah je potrebno zmanjšati na minimum, kar pomeni, da se ti podatki hranijo le v tolikšni meri in na tistih mestih, kot je to nujno potrebno. Če jih ne potrebujemo (več), jih je potrebno odstraniti. V vsakem primeru pa se nikoli ne sme hraniti podatkov z magnetnega traku kartice ali tri mestne številke na zadnji strani kartice.
- Enkriptirati je potrebno številke kartic (PAN – *personal account number*) in sicer z močnim enkriptirnim algoritmom, hkrati pa je potrebno zagotoviti tako upravljanje z dostopi do ključev teh šifrirnih algoritmov, da je možnost zlorabe in neavtoriziranih dostopov čim manjša.
- Zagotavljati je potrebno redno in učinkovito posodabljanje popravkov, s katerimi odpravljamo morebitne varnostne luknje v podatkovnih bazah. Ker so študije pokazale, da je tovrstno nameščanje popravkov s strani nekaterih DB administratorjev izredno redko, PCI standard zahteva, da so popravki nameščeni v enem mesecu od njihove uradne izdaje.
- Razvite spletne aplikacije morajo biti varne, brez varnostnih ranljivosti, ki bi omogočale zlorabe in napade na podatkovne baze, kot je SQL vrivanje (*SQL injection*). PCI nalaga administratorjem podatkovnih baz, da razvijalce spodbujajo in opozarjajo, da morajo kodo napisati tako, da v čim večji meri preprečijo možne ranljivosti.
- Upravljanje z uporabniškimi računi mora biti varno. Osnovno je seveda imeti uporabniške račune z močnimi gesli, poleg tega pa je potrebno imeti razdelano upravljanje z uporabniškimi vlogami in pravicami s ciljem, da se omeji dostop samo za točno določene namene in točno določene uporabnike, ki še zagotavljajo tisto, kar uporabniki pri svojem delu potrebujejo.
- Vse aktivnosti v podatkovnih bazah je potrebno zabeležiti (logirati). PCI pri beleženju zahteva najmanj sledeče podatke: uporabniško ime, tip dogodka, čas in datum in ostale tehnične informacije o vsakem posameznem dostopu določenega uporabnika do podatkov o imetnikih plačilnih kartic, aktivnostih administratorja in morebitnih neuspelih poskusih avtentikacije.

4.2.2 Revizijska sled v podatkovnih bazah

4.2.2.1 *Opredelitev revizijske sledi*

Revizijsko sled (angl. *audit log*) bom v nadaljevanju opredelil z dveh vidikov, z, recimo mu tako, strokovnega in internega vidika podjetja, ki ima tudi več praktične vsebine.

Zbiranje podatkov za revizijsko sled (angl. *auditing*) je zapisovanje in analiza dogodkov ali statistike za podajanje informacij o uporabi in zmogljivosti sistema, podani v jasni in razumljivi obliki [Bishop 2003 v [43]]. Cilj zagotavljanja revizijske sledi je ugotoviti, ali so kršene varnost in druge politike na sistemu [43]; te politike vključujejo tako splošne varnostne zahteve kot tudi specifične varnostne zahteve, kot jih podajajo zakonske določbe ali, kot na primer v primeru za to magistrsko nalogo, uredbe o varovanju podatkov o imetnikih plačilnih kartic.

V podjetju je revizijska sled opredeljena kot zbir vseh podatkov, ki identificirajo čas, vrsto, vsebino, identifikacijo podatkovnega sklopa, izvajalca in način spremembe podatkov oziroma spremembe načina obdelave podatkov, pri čemer se jo opredeljuje tudi kot orodje revizorjev, kontrolorjev in linijskih vodij.

Kot je navedeno v internem dokumentu podjetja, med drugim revizijska sled zagotavlja [16]:

- sledljivost in nepotvorljivosti podatkov,
- dokumentiranost okolja, v katerem so transakcije nastale,

- predstavitev zgodovinskega zaporedja transakcij o poslovnem dogodku,
- nadzor nad izvedbami kontrol,
- usklajenost z zakonodajo,
- možnost ugotavljanja pooblaščenosti izvedene spremembe,
- možnost ugotavljanja okoliščin varnostnega dogodka.

V skladu s PCI standardom interna politika opredeljuje še naslednje. Sled se navadno generira samodejno, ko se zgodi transakcija, določena aktivnost, dostop, procesiranje, pri čemer naj se to zgodi kar se da hitro, ko se zgodijo zgoraj omenjeni dogodki. Namenjena je rekonstrukciji dogodkov, s čimer je možna izsledljivost postopkov znotraj delovnega ali poslovnega procesa. Za revizijske sledi je značilno, da zahtevajo najvišjo stopnjo varovanja. Pravico do pregledovanja in raziskovanja revizijskih sledi imajo le določeni. Prav tako so vsi posegi, kot so reorganizacija, agregiranje ali kriptiranje, omejeni. Brisanje ni dovoljeno, razen po prenosu sledi v arhiv ali po koncu dobe hranjenja revizijske sledi. Pri opredeljevanju na občutljivih podatkih je pomembno opredeliti tudi rok hranjenja revizijskih sledi. Starejše zapise se navadno arhivira.

Fizično je poleg shranjevanja revizijske sledi v podatkovnih bazah le-te možno shranjevati tudi v datoteke. Spodaj je primer take datoteke (angl. *audit log file*). Zapisovanje v log datoteke je na primer uporabno pri beleženju dostopa do posameznih strani spletne aplikacije.

```
\AppName#:2012-04-03 12:35:10.212#:ca90217f-2b56-4b12-9a79-2a262ebecb8b#:domain\user1()#:p_Login#:G#:
\AppName#:2012-04-03 12:35:10.228#:bce1d7af-65ee-45e0-94e3-
4af8619c1c7b#:domain\user1()#:p_default#:G#:/en/default/
\AppName#:2012-04-03 12:35:10.541#:9fc230de-7114-4bca-b5e1-
be9f549b8a64#:domain\user1()#:p_KSA#:G#:/KSA.aspx
\AppName#:2012-04-03 12:35:32.228#:bf8d8a5b-bfa9-4c60-80df-
c9d89b355d44#:domain\user1()#:p_Search#:G#:/en/S/Search/
\AppName#:2012-04-03 12:35:33.275#:c62febd4-bd77-4ae7-8912-
d18a05ade41b#:domain\user1()#:p_KSA#:G#:/KSA.aspx
\AppName#:2012-04-03 12:51:47.041#:705dfcd8-a6ba-43e8-8767-
a7ab3ab7881b#:domain\user2()#:p_KSA#:G#:/KSA.aspx?q=333345654653534
\AppName#:2012-04-03 12:54:33.337#:266bca5b-fffd-4216-9074-
134f4c67477a#:domain\user1()#:p_KSA#:G#:/KSA.aspx?q=333345654653567
\AppName#:2012-04-03 12:57:55.744#:78ae331b-4f6d-4d61-9dfd-
c51f440b0aab#:domain\user2()#:p_ViewPage#:G#:/en/VP/ViewPage/-
|FormSubmitChB?on|cisChB?on|cmisChB?on|cnTB?|birthDateTB?__.__.____|fNameTB?s|lNameTB?k|companyTB?|taxN
oTB?|primaryKeyTB?|emsoTB?
\AppName#:2012-04-03 13:10:47.127#:b21ff88d-d0d1-405c-94f8-
91927b1e0188#:domain\user2()#:p_KSA#:G#:/KSA.aspx?q=333345654653534
\AppName#:2012-04-03 13:13:33.392#:8a741519-9a97-41db-b485-
8da2f2751ca6#:domain\user1()#:p_KSA#:G#:/KSA.aspx?q=333345654653546
\AppName#:2012-04-05 12:16:56.412#:4297d75a-6c7f-48ec-a78a-7d7924df56cd#:domain\user3()#:p_Login#:G#:
\AppName#:2012-04-05 12:16:56.709#:34bbe917-a1cc-4463-8d17-
ce848f157960#:domain\user3()#:p_Default#:G#:/en
\AppName#:2012-04-05 12:16:59.256#:2119f6e0-4fdd-4044-b618-
13a8c554ee12#:domain\user3()#:p_KSA#:G#:/KSA.aspx
\AppName#:2012-04-05 12:17:02.318#:3b8880fe-c14b-4400-ba9c-
c57a256e65b5#:domain\user3()#:p_Search#:G#:/en/S/Search/
\AppName#:2012-04-05 12:17:03.521#:8ce1538c-b782-495a-8bcc-
ea0d8032125b#:domain\user3()#:p_KSA#:G#:/KSA.aspx
\AppName#:2012-04-05 12:17:27.334#:2d0e9aec-5324-4b29-9a4b-
65275927df26#:domain\user3()#:p_SearchCounter#:G#:/en/S/SearchCounter/
\AppName#:2012-04-05 12:17:27.771#:0a31f10a-9e97-4efb-be58-
efb74c6b68fe#:domain\user3()#:p_KSA#:G#:/KSA.aspx
\AppName#:2012-04-05 12:17:29.177#:0161820e-976a-4d84-93da-
30b7d9788c42#:domain\user3()#:p_DocPage#:G#:/en/DP/DocPage/-
|cnTB?|amountFromTB?|amountToTB?|dateFromTB?05.04.2012|timeFromTB?00:00:00|dateToTB?05.04.2012|timeToTB
?23:59:59|GridSorting?TERM_DESC|PageSize?25
\AppName#:2012-04-05 12:17:29.865#:ea07f5f9-4a58-44df-8897-
ce27f6fe0a52#:domain\user3()#:p_ViewPage#:G#:/en/VP/ViewPage/-
```

```
|cnTB?|amountFromTB?|amountToTB?|dateFromTB?05.04.2012|timeFromTB?00:00:00|dateToTB?05.04.2012|timeToTB
?23:59:59|GridSorting?TERM_DESC|PageSize?25
\AppName#:2012-04-05 12:17:30.881#:252443b6-7685-41f3-a651-
9e6f63b03e15#:domain\user3()#:p_KSA#:G#:/KSA.aspx
```

Slika 10 - Primer zapisa revizijske sledi v datoteko (audit trail)

Spodaj je primer zapisa sledi v podatkovni bazi s pregledom v grafičnem uporabniškem vmesniku (angl. *GUI – graphical user interface*).

Audit Process Log

Start Date: 2009-12-08 10:57:50 End Date: 2009-12-08 11:57:50
Aliases: ON

Audit Process Log Id	Login Name	Run Id	Timestamp	Audit Process Id	Audit Process Description	Audit Task Id	Audit Task Description	Event Type	DETAIL
23		3	2009-12-08 11:51:48.0	1000000	Weekly database changes	0		process stop	Finished manu
22		3	2009-12-08 11:51:48.0	1000000	Weekly database changes	0		deliver	Result(s) dist
21	admin	3	2009-12-08 11:51:48.0	1000000	Weekly database changes	1000002	event status transition bp	task stop	Finish proces
20	admin	3	2009-12-08 11:51:48.0	1000000	Weekly database changes	1000002	event status transition bp	task start	Start audit tas
19	admin	3	2009-12-08 11:51:48.0	1000000	Weekly database changes	1000001	outstanding events bp	task stop	Finish proces
18	admin	3	2009-12-08 11:51:48.0	1000000	Weekly database changes	1000001	outstanding events bp	task start	Start audit tas
17	admin	3	2009-12-08 11:51:48.0	1000000	Weekly database changes	1000000	failed logins	task stop	Finish proces
16	admin	3	2009-12-08 11:51:48.0	1000000	Weekly database changes	1000000	failed logins	task start	Start audit tas
15		0	2009-12-08 11:51:48.0	1000000	Weekly database changes	0		process start	Start manual a changes

Records 1 to 9 of 9

Slika 11 - Primer poročila/zapisa revizijske sledi v podatkovni bazi [44]. Primer prikazuje podrobnejši zapis aktivnosti z najpomembnejšimi podatki, kot so čas/datum, uporabniško ime, opis aktivnosti.

4.2.2.2 *DAM - Database Activity Monitoring sistem*

4.2.2.2.1 Opredelitev DAM

DAM ali *database activity monitoring* (lahko tudi DAS – *database auditing system*) je sistem, ki zajema in zabeleži vse SQL dogodke v realnem času ali skoraj realnem času, vključujoč administratorske aktivnosti, pri čemer zna generirati opozorila o kršenju varnostne (ali drugačne) politike združbe. DAM rešitev je lahko tako rešitev v obliki strojne opreme, kot tudi čisto programske opreme, največkrat pa kombinacija obojega. Čeprav obstaja veliko produktov, ki beležijo aktivnosti, je po definiciji DAM sistem, če ima možnost [45]:

- neodvisnega nadzora in beleženja vseh aktivnosti na podatkovnih bazah, vključno z administratorskimi aktivnostmi in beleženjem SELECT stavkov. Orodja so zmožna beleženja vseh vrst SQL transakcij: DML, DDL, DCL, TCL[15] aktivnosti,
- varno shraniti v prejšnji alineji omenjene aktivnosti zunaj predmetnih podatkovnih baz,

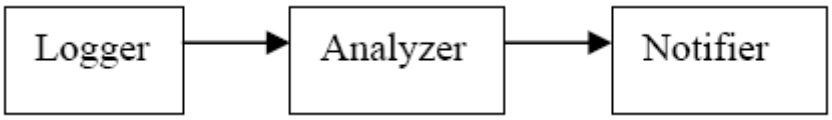
[15] Okrajšave skupin oziroma vrst ukazov v SQL žargonu:
DQL: *Data Query Language*, DML: *Data Manipulation Language*, DDL: *Data Definition Language*, TCL: *Transaction Control Language*, DCL: *Data Control Language*
DQL: SELECT
DML: DELETE, INSERT, UPDATE
DDL: CREATE, DROP, TRUNCATE, ALTER
TCL: COMMIT, ROLLBACK, SAVEPOINT
DCL: GRANT, REVOKE

- agregirane in korelirane aktivnosti iz več heterogenih sistemov za upravljanje podatkovnih baz, kot so Microsoft SQL Server, Oracle DBMS, IBM DB2 in druge, hkrati pa kljub različnostim v SQL jezikih znajo normalizirati transakcije,
- razdelitve nalog na administratorje podatkovnih baz; kot že omenjeno, mora imeti možnost preprečitve dela z logi in zapisi revizijskih sledi, administratorji prav tako ne smejo delati z DAMom,
- generiranja opozoril glede na kršenje politike, opredeljene v DAM; torej DAM po definiciji ne beleži samo aktivnosti, ampak omogoča *real-time* nadzor in opozorila na podlagi pravil. Za slednje so v [45] navedli preprost primer pravila, ki se glasi: V primeru poizvedbe na bazi, pri kateri so navzoče informacije o kreditni kartici in rezultat vrne več kot eno vrstico, se sproži opozorilo.

Poglejmo tudi Gartnerjevo opredelitev [44]. Po njej DAM sistemi uporabljajo različne mehanizme za zbiranje podatkov, jih združujejo v neki centralni enoti za analizo in poročajo o dogodkih, ki kršijo vnaprej opredeljene politike, ali o zaznanih anomalijah. Bistvena uporabnost DAM sistemov je po Gartnerju funkcionalnost nadzora uporabnikov s privilegiranimi pravicami; druga glavna funkcionalnost pa je pomoč upravljalcu za varnost pri nadzoru dostopa do podatkovnih baz z občutljivimi podatki. Gartner v svoji definiciji še navaja, da se funkcionalnosti DAM sistemov vse bolj širijo in dodajajo nove; primer je zmožnost generiranja opozoril pri zlonamernih dejavnostih ali zmožnost zaznave neprimernega ali neodobrenega dostopa administratorjev do podatkovnih baz.

4.2.2.2.2 Arhitektura DAM

Obe opredelitvi, tako Gartnerjeva kot tista iz članka Richa Mogulla [45], sta si zelo podobni. Sistem beleži, analizira dogodke in obvlada opozorila pri poizvedbah v podatkovno bazo, kar nakazuje tudi čisto osnovna arhitektura značilnega DAM sistema, ki ga orisuje slika 12.



Slika 12 - Osnovna arhitektura DAM sistema
Vir: (Bishop 2003 v [43])

Zapisovalnik (*logger*) zbira podatke na podlagi varnostne, državne ali kakšne druge politike. Analizator (*analyzer*) kot vhod vzame v zapisovalniku zbrane log zapise, kjer ugotovi, ali se je nek dogodek, ki nas zanima, zgodil, ali če je potrebno zabeležiti še kak podatek. Rezultate analize prejme javljalnik (*notifier*), ki jih sporoča entitetam, ki se zanimajo za rezultate. Te entitete so lahko ljudje, računalnik ali kaka druga naprava.

V podjetjih, kjer hranijo ogromno podatkov in kjer je opravljenih veliko število poizvedb v podatkovnih bazah, je ročno analiziranje morebitnih zlorab seveda praktično nemogoče. V ta namen, kot je že bilo omenjeno, poznajo naprednejši sistemi DAM možnost avtomatske zaznave zlonamernih dogodkov na podlagi določenih metod, ki v osnovi temeljijo na dveh načinih delovanja [46]:

- prepoznavanje znanih vzorcev – pri tem načinu se išče znane vzorce pri izvajanju ukazov, ki nakazujejo napade;
- zaznavanje anomalij – pri tem načinu se išče razlike v ukazih glede na prejšnje, iz zgodovine poznane varne ukaze.

Po ocenah Schonlaua in drugih [46], so najbolj močni pristopi za odkrivanje vdorov in ranljivosti tisti, ki temeljijo na drugi, zgoraj navedeni alineji – ukazi, ki v preteklosti niso bili zaznani, lahko

nakazujejo poskuse zlorab. Za primer enega od takih mehanizmov bom na kratko opisal mehanizem, imenovan MDAD – *Malicious Data Access Detector* [46], ki dela vzporedno s podatkovno bazo in podatke analizira v realnem času. To torej pomeni, da se v primeru zlorabe ali napada takoj poda obvestilo administratorju podatkovne baze, npr. s pošiljanjem elektronske pošte ali sms sporočila. Taki mehanizmi v osnovi potekajo v dveh fazah: učenju in zaznavi. Najprej k prvi fazi – učenju. Sistem mora biti konfiguriran tako, da beleži vse operacije nad podatki (*select*, *insert*, *update*, *delete*) – s tem se kreira revizijska sled. Na podlagi zbranih podatkov se naredi graf profilov transakcij, ki kasneje služijo za odkrivanje zlorab. Pravzaprav gre za bazo pravil in profilov, ko jih sistem za odkrivanje uporablja pri svojem delovanju.

DAM mora delovati neodvisno od DBMS (*Database Management System*) sistema, torej neodvisno od podatkovne baze.

4.2.2.2.3 Druge uporabe in izdelki

Kot zanimivost lahko omenim še to, da se poleg podpore za standard PCI DSS, DAM uporablja tudi za druge certifikacije, kot je na primer podpora Sarbanes-Oxley Act (SOX)[16] ali *Health Insurance Portability and Accountability Act* (HIPAA)[17].

Naj na koncu poglavja omenim še nekaj izdelkov, ki so trenutno na trgu, specializiranih za rešitve na področju beleženja revizijskih sledi in analiziranja dostopov do občutljivih podatkov, ti so: McAfee Database Activity Monitoring, IBM Guardium, Imperva SecureSphere, Secerno, Tizor, Sentrigo, Idera itd.

4.2.3 Implementacija beleženja revizijskih sledi v skladu s standardom PCI z DAM sistemi

Kot že večkrat povedano, vključuje PCI DSS standard množico zahtev, v nadaljevanju pa se bom seveda omejil na nekaj ključnih, ki se nanašajo na varnost podatkovnih baz. Pregledal jih bom po točkah, kot so navedene v zadnji specifikaciji standarda (verzija 3.1) [39]. Za vsako od točk bom nato navedel tehnološki koncept rešitve, ki naj bi jih ponujali proizvajalci DAM.

PCI DSS poglavje 2.2

Ta razdelek zahteva, da imajo združbe, ki želijo izpolniti PCI, pripravljene konfiguracijske standarde za vse komponente (med drugim tudi podatkovne baze), ki vključujejo vse varnostne ranljivosti in so v skladu s standardi, ki veljajo za industrijo procesiranja plačilnih kartic. Primeri takih standardov so *SysAdmin Audit Network Security Network (SANS)*, *National Institute of Standards Technology (NIST)*, in *Center for Internet Security (CIS)*.

Ker so znane tako pomanjkljivosti podatkovnih baz kot tudi načini, kako baze konfigurirati, da se varnostna tveganja in ranljivosti zmanjšajo, morajo administratorji podatkovnih baz poskrbeti, da se jih pokrije (tudi) z uporabo zgoraj naštetih, dobro dokumentiranih standardov.

Koncept rešitve:

DAM imajo vgrajene politike, ki preverjajo nastavitve podatkovne baze glede na dobre varnostne rešitve, ki so jih postavili že prej omenjeni CIS in NIST SRR (Security Readiness Review for SQL Server) pa tudi varnostne rešitve, ki jih postavlja interna varnostna politika. Večino DAM rešitev je torej možno konfigurirati tako, da upoštevajo (tudi) politiko podjetja.

PCI DSS poglavje 7.1 in 7.2

V teh dveh razdelkih je zahtevano, da združbe omejijo dostop do računalniških virov in podatkov o imetnikih plačilnih kartic samo tistim posameznikom, ki potrebujejo tak dostop. Potreben je tak

[16]SOX ali Sarbanes-Oxley Act je ameriški zvezni zakon, ki uvaja strožje upravljanje, računovodstvo in finančno poročanje ameriških javnih podjetij.
[17]HIPAA je ameriški zakon, ki ureja področje varovanja pacientovih kartonov in drugih zdravstvenih podatkov.

mehanizem (za vse sisteme), ki privzeto zavrne dostop za vse (angl. *deny all users*), potem pa se dostopi omogočajo posamezno in le toliko, kolikor je potrebno.

Združba mora ustvariti politiko, ki jasno določi, kdo ima pravice in dostope do občutljivih podatkov. Administratorji podatkovnih baz morajo izpolniti oziroma izvesti zahteve, ki jih narekuje omenjena politika združbe.

Koncept rešitve:

DAM sistemi omogočajo vpogled v dodeljene in dedovane pravice na tabelah, ki vsebujejo občutljive podatke, s čimer je omogočen takojšen nadzor nad tem, da imajo pravice dostopa do podatkov iz teh tabel samo tisti, ki jih morajo imeti oziroma za to obstaja poslovni razlog. Prav tako večina rešitev omogoča analizo dodelitev uporabnikov in vlog, s čimer je hitro mogoče za določenega uporabnika/vlogo dejansko ugotoviti, kakšna dovoljenja ima. Podatki so torej agregirani. S tem je olajšano ročno preverjanje, ki bi ga morali opraviti ljudje.

PCI DSS poglavje 8.1 (8.1.4) in 8.5

Ti dve poglavji govorita o zahtevi, da morajo vsi uporabniki v združbi imeti svoje lastno uporabniško ime in geslo, s katerim se identificirajo v sisteme z občutljivo vsebino. Pri tem skupinska, deljena in generična uporabniška imena niso dovoljena. Glede na razdelek 8.1.4 morajo biti uporabniški računi, ki so neaktivni 90 dni, onemogočeni. V tem primeru morajo biti dostopne pravice takih računov ukinjene.

Administratorji morajo upoštevati vse uporabnike, ki imajo dostop do podatkovnih baz. Uporabniški računi morajo biti našteti, morajo biti identificirani in za vsak račun mora biti preverjena istovetnost lastnika računa. Tako je zagotovljena posameznikova odgovornost za vsa dejanja, ki jih naredi na bazi, hkrati pa je zagotovljena tudi učinkovitost zapisov revizijske sledi za vsakega posameznika. V primeru suma na kakršnokoli zlorabo bo odkrivanje vpletenih in njihovih dejanj bistveno olajšano.

Koncept rešitve:

DAM omogočajo številne preglede uporabnikov in skupin iz različnih zbirk uporabnikov. S tem lahko pridobimo seznam uporabnikov, ki imajo dostop do občutljivih podatkov, pa jih morda ne bi smeli imeti.

PCI DSS poglavje 10.1

Poglavje 10.1 od združbe zahteva, da le-te uredijo proces za povezavo vseh dostopov do sistemskih komponent, še posebej dostopov z administratorskimi privilegiji, za vsakega posameznega uporabnika.

Podobno kot v poglavjih 8.1 in 8.5 je tukaj pomembno ustvarjanje revizijske sledi, torej zapisovanje dogodkov in dejanj, ki jih dobimo s povezavo dostopa do komponent podatkovne baze s posameznim uporabnikom, še posebej z uporabnikom s privilegiranim dostopom. Kot že rečeno, je to koristno za odkrivanje in raziskavo dejavnosti po odkritem incidentu.

Koncept rešitve:

Za pokritje te točke rešitve seveda beležijo vse aktivnosti v podatkovnih bazah. Rešitve naj bi podatke zbirale in shranjevale učinkovito in varno. Na voljo so orodja za analizo in poročila, tudi za forenzična poročila. Orodja vsebujejo lahko tudi več funkcionalnosti za zaščito in varnost podatkov, kot tudi metod za pregled dogodkov brez da bi izpostavljali informacije o uporabniških računih.

PCI DSS poglavje 10.2 in 10.3

Ti poglavji zahtevata avtomatsko ustvarjanje revizijske sledi za vse sistemske komponente, s čimer lahko rekonstruiramo število ključnih dostopov do podatkov o imetnikih plačilnih kartic in sicer, kakor je navedeno v podrazdelkih:

- 10.2.1 vse posameznikove dostope do podatkov
- 10.2.2 vse posameznikove akcije z administrativnimi in privilegiranimi pravicami
- 10.2.3 dostope do vseh revizijskih sledi
- 10.2.4 vse napačne logične poskuse dostopa
- 10.2.5 uporabo identifikacijskih in avtentikacijskih mehanizmov
- 10.2.6 inicializacijo revizijskih sledi
- 10.2.7 kreiranja in brisanja objektov na sistemskem nivoju

Razdelek 10.3 zahteva, da so za vsako sistemsko komponento v revizijski sledi zabeleženi naslednji podatki:

- 10.3.1 uporabnikova identifikacija
- 10.3.2 vrsta dogodka
- 10.3.3 datum in čas
- 10.3.4 informacija o uspehu ali zavrnitvi
- 10.3.5 izvor dogodka
- 10.3.6 identiteta oziroma ime prizadetega podatka, sistemske komponente ali drugega vira

Koncept rešitve:

Rešitve morajo beležiti vse dogodke iz sekcije 10.2 in vse podrobnosti, zahtevane v sekciji 10.3. Vsebuje sistem za opozarjanje, po možnosti tudi števce in grafično predstavitev trenda za identifikacijo anomalij pri aktivnostih, pogosto povezanimi s sumljivimi aktivnostmi.

PCI DSS poglavje 10.5

Revizijske sledi ne smejo biti spremenjene, je zapisano v tem poglavju. Nekdo, ki vdre v omrežje, bo lahko želel spremeniti revizijsko sled, s katero bi prekril svoje aktivnosti. Torej, brez primerne zaščite revizijskih podatkov se njihova vseobsežnost, kakovost, točnost in integriteta ne more zagotavljati, kar posledično pomeni, da je uporabnost podatkov in orodij za delo z njimi v primeru kompromiranja nična.

Koncept rešitve:

Glede na zgornji odstavek morajo izdelovalci DAM seveda poskrbeti za nespremenljivost (*imutabilnost*) zabeleženih podatkov, vključno z zabeleženimi aktivnosti administratorjev. Vse spremembe logov morajo biti zaznane, prav tako pa je zaželeno, da imajo sistem za opozarjanje ustreznim osebam za ukrepanje.

PCI DSS poglavje 10.7

V tem poglavju je podana zahteva, da morajo podjetja hraniti revizijsko sled vsaj eno leto, pri čemer mora biti le-ta takoj dostopna znotraj treh mesecev. Ostalo gre lahko v arhiv.

Koncept rešitve:

Sistem shranjuje podatke v centralni repozitorij, ki omogoča enostavno arhiviranje podatkov za željeno obdobje.

V zadnjem poglavju bom opisal primer iz prakse s poudarkom na opisu postopka izbire rešitve. Čeprav sem v zaključnem poglavju želel podrobneje opisati, kako in na kakšen način ter kateri produkt smo

izbrali, sem bil zaradi zaupnosti informacij primoran besedilo oblikovati tako, da ne bi razkril podatkov zaupne narave. Tako v nadaljevanju ne operiram s konkretnimi primeri izdelkov in njihovih proizvajalcev, ampak se osredotočam predvsem na postopek izbire. Konkretne izdelke, ki so trenutno na trgu in ki so tudi prišli v krog izdelkov, potencialno zanimivih tudi za nas, sem naštel že v razdelku 4.2.2.2.3.

4.2.4 Potek izbire DAM rešitve

Ker je seveda DAM rešitev za podjetje večja investicija (predvsem na račun cene nakupa in vzdrževanja ter ostalih licenčnin), je bilo potrebno dosledno opredeliti korake izbora in kriterije, ki so narekovali končni izbor. Izbor je tako potekal v naslednjih osmih fazah:

1. Izbor zahtev iz standarda PCI, ki jih mora izdelek pokrivati.
2. Izbira širšega kroga izdelkov s seznama, narejenega preko informacij, zbranih na spletnih straneh proizvajalcev, ki podpirajo zahteve iz točke 1.
3. Opredelitev in izdelava spiska internih zahtev.
4. Izbira največ štirih izdelkov v finalni izbor.
5. Podrobnejša analiza in testiranje vsakega od izbranih izdelkov iz izbora iz točke 3.
6. Analiza rezultatov in testiranj.
7. Predstavitev rezultatov vodstvu podjetja.
8. Končna odločitev vodstva podjetja v sodelovanju z zaposlenimi v oddelku, ki je pripravil rezultate iz točke 7.

V nadaljevanju bom podrobneje opisal vsako od zgoraj navedenih točk.

V prvi fazi je bilo torej potrebno zbrati zahteve in pripraviti seznam vseh PCI zahtev, ki jih mora produkt podpirati. Ta seznam, ki sem ga sestavil in podrobno opisal v razdelku 4.2.3, je osnova poizvedovanja za izdelki, ki so prišli v širši izbor. Velika večina proizvajalcev ima na svojih spletnih straneh ob predstavitvi izdelka tudi zapis o podpori PCI standardu. Navadno je že iz tega vidno, ali izdelek v skladu s podanimi zahtevami iz PCI standarda zadostuje našim in predvsem PCI potrebam. Na podlagi tega je bilo v podjetju najprej izbranih nekaj rešitev, ki so bili uvrščeni na širši seznam produktov, potencialno primernih za pokritje PCI zahtev. Ta seznam je obsegal osem izdelkov.

Sledila je priprava vseh internih zahtev, ki so bili sestavljeni na podlagi sestankov z vsemi vpletenimi zaposlenimi: sodelavci iz službe za informacijsko zaščito, ki vodijo projekt, sistemskimi administratorji, razvijalci informacijskih sistemov in podatkovnih baz in seveda administratorji podatkovnih baz. Povzetek zahtev je bil strnjen v naslednjih točkah, pri čemer je prvih sedem točk odločilnega pomena (angl. *deal breaker*):

1. Podpora za MS-SQL 2005 in 2008 RC2 podatkovno bazo, tako 32-bit kot 64-bit verzijo.
2. Podpora ostalim DBMS sistemom, kot so Oracle DB, IBM DB2 itd.
3. Beleženje revizijskih sledi standardnih DML (*insert*/*update*/*delete*) dogodkov.
4. Beleženje revizijskih sledi poizvedb, DQL (*select*-ov), shranjenih procedur, T-SQL izvedb dogodkov.
5. Nastavitve za filtriranje po tabeli, uporabniku, shranjenih procedurah itn.
6. Filtriranje določenih revizijskih sledi glede na izvor. Kot primer lahko navedem replikacije v realnem času, za katere ne želimo, da se jih beleži, ker bi to pomenilo povečanje performančne porabe.
7. Zasnova nekaterih aplikacij je takšna, da se iz aplikativnega strežnika do podatkovne baze dostopa le z enim SQL uporabniškim računom. Kadar aplikacijo (npr. *gui odjemalca*) uporablja več uporabnikov, se informacija o dejanskem uporabniku, ki je naredil določeno poizvedbo iz odjemalca, zabriše, saj imamo na aplikativnem strežniku le en račun, ki dostopa

do baze. V Microsoftovih okoljih je za ta problem lahko rešitev uporaba CONTEXT_INFO[18], s katerim lahko vsako transakcijo v podatkovno bazo opremimo z enoličnim identifikatorjem, ki ga zabeležimo tudi pri uporabniški zahtevi iz odjemalca na aplikativni strežnik ter tako zagotovimo uparjanje zahteve, ki je prišla z odjemalca s pripadajočo zahtevo, ki poizveduje v bazi. Tako imamo podatek, kateri dejanski uporabnik je naredil določeno transakcijo v bazi, vedno na voljo. Najti moramo le par zahteva na strežnik – transakcija v bazi glede na ključ, npr. guid. V izbranem DAM je torej potrebna podpora za grupiranje dogodkov (*events*), transakcij (*transactions*), izvedb (*executions*), poizvedb (*selects*) z uporabo CONTEXT_INFO. Kot se je kasneje izkazalo, je bila to ena od ključnih točk pri vrednotenju rezultatov.

8. Majhna poraba računalniških resursov (virov) v primeru strežniških procesov, še sprejemljiva zgornja meja je 5%.
9. (Z)možnost dostopa iz po meri narejene (*custom*) GUI aplikacije, konzolne aplikacije preko API-ja ali kakšnega drugega načina.
10. Arhiviranje podatkov in obnova podatkov iz arhivov.
11. Dostop do uporabniških in časovnih žigov, informaciji o aplikaciji, tipu transakcij, starih in novih podatkih (kjer je to možno) časovni žigi revizij, časovni žigi brisanj.

Sledila je naslednja točka, v kateri so bili poiskani kontakti (naslovi elektronske pošte) proizvajalcev. Na podlagi zgoraj omenjenih kriterijev oziroma zahtev, je bilo sestavljeno povpraševanje, ki je bilo vsem proizvajalcem poslano v enaki obliki. Morali so odgovoriti, kaj od naštetih funkcionalnosti podpirajo. Poleg tega je bila želja po dodatnih informacijah o optimalnih konfiguracijah strojne in programske opreme, zmogljivosti ter informacije o ceni in licencah. Odgovori so bili v doglednem času dobljeni od vseh proizvajalcev, nekateri bolj, drugi manj izčrpni.

Na podlagi prejetih odgovorov in analize sta bila v finalni izbor izbrana le dva izdelka, ki sta bila v nadaljevanju preizkušana v testnem okolju z namenom preizkusa rešitve v praksi. Sestavljen je bil urnik predstavitev, namestitev, konfiguracij ter demonstracij izdelkov. Namestitvi strojne in programske opreme sta potekali zaporedno, izvajali so jih proizvajalčevi ljudje v sodelovanju s sistemskimi administratorji in DB administratorji. Vsaka od obeh rešitev je bila v testu približno dva meseca.

Rezultat testiranj je bil dokument v obliki poročila za vodstvo združbe, ki je odločalo, kateri od izdelkov je bil izbran.

4.3 PCI standard in razvoj varnih aplikacij

Drug primer zaznanega varnostnega tveganja, ki sem ga identificiral v razdelku 3.4.6.2, je poskus napada z SQL vrivanjem, ki je seveda le eden od (najpogostejših) primerov napadov iz področja nepravilnega/ne-varnega razvoja programske opreme. PCI DSS standard namenja celotno poglavje/sklop zahtev s področja varnih sistemov in programske opreme in sicer Zahteva 6 - Razvijati in vzdrževati je potrebno varne sisteme in aplikacije. Ker bi se želel v nadaljevanju ozko omejiti zgolj na sam razvoj, so v nadaljevanju podrobneje identificirane zahteve standarda PCI DSS za varno

[18]CONTEXT_INFO je Microsoft SQL Server funkcionalnost, s katerim lahko dodatno opišemo kontekst posameznega SQL stavka, ki ga naredimo v bazi. Primer uporabe CONTEXT_INFO v SQL [47]:

```
SET CONTEXT_INFO 0x1256698456
GO
SELECT CONTEXT_INFO()
GO
```

kodiranje in razvoj varnih aplikacij. Prav tako nekoliko podrobneje opisujem razdelek PCI 6.5.1, ki je namenjen ranljivosti vrivanja, ki je bila kot primer primarno identificirana ranljivost.

4.3.1 Zahteve, ki jih ponuja PCI za razvoj varne programske opreme

Zahtev, ki jih PCI DSS namenja varnim sistemom, je mnogo. V nadaljevanju sem izbral le tiste, ki so neposredno povezane z varnim kodiranjem, katerega primer je obramba pred najpogostejšim napadom (in tudi enim najnevarnejših) na spletne aplikacije, SQL vrivanjem. Vsako od izbranih poglavij sem povzel le na kratko, izjema je le poglavje 6.5.1, ki opisuje ranljivost (SQL) vrivanja. S tako identificiranimi zahtevami bom na primerih opisal, kako smo v razvojnem oddelku izboljšali kakovost in varnost razvitih aplikacij.

PCI DSS poglavje 6.3

Ta razdelek zahteva razvoj tako internih kot zunanjih programskih rešitev na način, ki je v skladu s PCI DSS, temelji na najboljših praksah industrije razvoja programske opreme in zahteva vpletenost informacijske varnosti čez celoten življenjski cikel razvoja. Če se ne upošteva varnosti med fazami opredelitve zahtev, načrtovanja, analize in testiranja, potem se lahko ranljivosti hote ali nehote pojavijo v produkcijskem okolju.

PCI DSS poglavje 6.3.2

Poglavje 6.3.2 govori o aktivnosti pregleda kode (*code-review*) pred prehodom v produkcijo, s katero se (lahko) prepozna morebitne škodljive dele slabo napisane programske kode. Za ta namen se lahko uporabi tako avtomatske kot ročne postopke, pri čemer je potrebno upoštevati, da kodne spremembe pregleda posameznik, ki ni avtor te kode, pri čemer ta posameznik pozna tehnike pregleda kode in prakse varnega kodiranja; nadalje morajo biti popravki v skladu z navodili varnega kodiranja, popravki morajo biti implementirani pred kakršnim koli prehodom v produkcijo, pri čemer jih potrdi vodja razvoja.

PCI DSS poglavje 6.5

Ta razdelek zahteva, da naj se razvijalci, da bi preprečili najbolj pogoste ranljivosti v kodi v procesu razvoja, učijo in izobražujejo s področja tehnik varnega programiranja (predvsem kako se izogniti najbolj pogostim kodnim ranljivostim in kako so občutljivi podatki shranjeni v računalniškem spominu) ter razvijajo aplikacije na podlagi navodil varnega programiranja.

PCI DSS poglavje 6.5.1

V tem razdelku standard za ranljivosti vrivanja navaja, da je potrebno preučiti politike razvoja in postopke razvoja ter intervjuvati odgovorne osebe za to, da se uporabi validacijo vhodnih uporabniških podatkov ter s tem prepreči spremembo pomena ukazov in poizvedovanj ter da se z uporabo parametriziranih poizvedovanj prepreči več vrst vrivanj (SQL, OS command, LDAP, XPath). V razlagi standard nadalje poda opis, kaj so vrivanja. Opisuje jih kot enega najpogostejših načinov za ogrožanje aplikacij. Vrivanje se zgodi, ko se podatki, podani s strani uporabnika, pošljejo kot del ukaza ali poizvedovanja pošljejo v interpreter.

PCI DSS poglavje 6.5.2 – 6.5.10

V teh razdelkih standard še za ostale najpogostejše ranljivosti navaja, da je potrebno preučiti politike razvoja in postopke razvoja ter intervjuvati odgovorne osebe za to, da se preveri, da tehnike kodiranja preprečujejo še naslednje grožnje: *buffer overflow* (6.5.2), nezadostna zaščita hrambe kriptografskih podatkov (6.5.3), nezadostna zaščita podatkov pri prenosu (6.5.4), neprimerno rokovanje oziroma obravnava napak (6.5.5), vse ranljivosti visokega tveganja kot jih identificira proces za odkrivanje ranljivosti (kot je opredeljeno v zahtevi PCI DSS 6.1) (6.5.6), podtikanje skript (*cross-site scripting* –

XSS) (6.5.7), nezavarovan neposreden dostop do objektov (6.5.8), potvarjanje spletnih zahtevkov (*cross-site request forgery – CSFR*) (6.5.9), napaka pri overjanju in upravljanju sej (6.5.10).

PCI DSS poglavje 6.6

Predvsem za javno dostopne spletne strani in spletne aplikacije je potrebno redno preverjati in slediti novim grožnjam in ranljivostim, hkrati pa je potrebno zagotoviti, da so te aplikacije zaščitene pred znanimi napadi na naslednje načine:

- Preverjanje javno dostopnih spletnih aplikacij z ročnim ali avtomatskimi orodji za oceno varnostnih tveganj vsaj enkrat letno ali ob večjih spremembah
- Namestitev avtomatskih tehničnih rešitev, ki zaznajo spletne napade (primer take rešitve je spletna požarna pregrada). Te tehnične rešitve je potrebno postaviti pred aplikacije; s tem je skeniran ves promet, ki gre do spletne aplikacije

4.3.2 Načini in primeri preprečevanja SQL vrivanj v kodi in podtikanja skript (XSS)

V oddelku pri razvoju spletne aplikacije za banke uporabljamo nekaj standardnih tehnik za preprečevanje možnosti vdora z SQL vrivanjem, kot jih navaja organizacija OWASP [48]:

- Uporaba parametriziranih stavkov
- Uporaba shranjenih procedur
- Preverjanje vseh vrednosti, ki jih lahko vnese uporabnik
- Uporaba minimalnih privilegijev uporabnikov podatkovnih baz, uporaba pogledov
- Seznam dovoljenih vhodnih parametrov

Z uporabo parametriziranih stavkov lahko podatkovna baza loči med kodo in podatki ne glede na to, kaj uporabnik vnese, saj morajo razvijalci pri uporabi parametriziranih procedur najprej napisati vso kodo in šele nato podati vsak posamezen parameter, ki nastopa v poizvedbi. V vsakem primeru je to najpomembnejša tehnika preprečevanja SQL vrivanj in edini način, s katerim mora biti pisana vsaka aplikacija, ki uporablja dostop do podatkovne baze; poleg tega je tak način pisanja stavkov preprostejši kot pisanje dinamičnih poizvedb.

Kot je zapisano v [48], s pripravljenimi stavki napadalec ne more spremeniti namena oziroma vsebine poizvedbe, tudi če le-ta ukaze uspe vnesti. V primeru kode na sliki 13 je najprej prebran *query string*[19] iz URL, v katerem je uporabnikovo uporabniško ime. Nato se s tem uporabniškim imenom naredi poizvedba o podatkih o tem uporabniku. Če v spodnjem primeru napadalec v URL vpiše Janez OR '1'='1', bo le-ta dobil podatke o vseh uporabnikih namesto le o Janezu. V primeru uporabe parametriziranega poizvedovanja pa se bo v pogoj vnesel cel zapis Janez OR '1'='1' kot parameter `Username` in tako poizvedba ne bo vrnila nič (Slika 14). V nadaljevanju je za demonstracijo prikazan izsek kode kot primer uporabe parametriziranih stavkov v jeziku C#.

```
txtUsername = getRequestString("Username");
sqlQuery = "SELECT * FROM Users WHERE Username = " + txtUsername;
```

Slika 13 - Primer kode, ji je podvržena SQL vrivanju

```
txtUsername = getRequestString("Username");
sqlQuery = "SELECT * FROM Users WHERE Username = "@0";
command = new SqlCommand(sql);
command.Parameters.AddWithValue("@0", txtUsername);
```

Slika 14 - Primer uporabe parametrizirane poizvedbe

[19] Niz znakov, ki predstavlja parametre v URL naslovu.

```
private static void UpdateDemographics(Int32 customerID,
    string demoXml, string connectionString)
{
    // Update the demographics for a store, which is stored
    // in an xml column.
    string commandText = "UPDATE Sales.Store SET Demographics = @demographics "
        + "WHERE CustomerID = @ID;";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlCommand command = new SqlCommand(commandText, connection);
        command.Parameters.Add("@ID", SqlDbType.Int);
        command.Parameters["@ID"].Value = customerID;

        // Use AddWithValue to assign Demographics.
        // SQL Server will implicitly convert strings into XML.
        command.Parameters.AddWithValue("@demographics", demoXml);

        try
        {
            connection.Open();
            Int32 rowsAffected = command.ExecuteNonQuery();
            Console.WriteLine("RowsAffected: {0}", rowsAffected);
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
}
```

Slika 15 - Primer uporabe parametriziranih stavkov v jeziku C#
Vir: [49]

4.4 PCI standard za področje penetracijskega varnostnega testiranja programske opreme

Zadnji razdelek predstavitve implementacije PCI DSS standarda v podjetje se navezuje na točko T1.3, manko varnostnega testiranja programskih rešitev. Za to tveganje sem v poglavju 3.4 tega magistrskega dela našel ublažitev v poglavju 11.3 (z razdelki 11.3.1, 11.3.2, 11.3.3 in 11.3.4) PCI DSS standarda.

Poglavje 11.3 PCI standarda je v celoti namenjen zahtevam s področja penetracijskega testiranja. Zajema 4 podpoglavja oziroma razdelke. Uvodoma je podanih nekaj alinej, v katerih je priporočeno, da združba udejanji metodologijo za penetracijsko testiranje, ki vključuje npr. to, da metodologija sloni na nekem pristopu, ki je že uveljavljen oziroma standardiziran, upošteva celoten obseg okolja o podatkih o imetnikih plačilnih kartic, vključuje testiranje tako zunaj kot znotraj omrežja, vključuje testiranje za potrditev kontrol segmentacije. Združba naj bi še opredelila penetracijske teste na aplikacijskem nivoju, ki naj zajema vsaj ranljivosti iz poglavja 6.5 standarda PCI DSS. O teh ranljivosti sem podrobneje že pisal v poglavju 4.3. Opredelila naj bi še penetracijske teste na mrežni ravni za mrežne komponente in operacijske sisteme, pregledala in razmislila o grožnjah in ranljivostih, zaznanih v zadnjih 12 mesecih ter opredelila rezultate penetracijskih testov in aktivnosti reševanja morebitnih zaznanih groženj in ranljivosti.

4.4.1 Udejanjenje penetracijskega testa za primer spletne aplikacije

Tako kot predvideva standard, se v združbi enkrat letno izvede penetracijsko testiranje s strani zunanjega izvajalca, ki se menja na vsake tri leta. Opravi se test vseh kritičnih sistemov in aplikacij, dostopnih iz zunanjega sveta. V nadaljevanju se bom za primer omejil na aplikacijo, ki sem jo z vsebinskega vidika podrobneje opisal v razdelku 3.4.6.2. in ki je, tudi že omenjeno, razvita in vzdrževana v oddelku.

Preden damo aplikacijo v penetracijsko testiranje zunanjemu izvajalcu, izvedemo tudi interno testiranje z namenskim orodjem. S tem orodjem se najprej v testirani aplikaciji posname scenarij testiranja, ki ni nič drugega kot vsebinska uporaba aplikacije. Ko je scenarij posnet, se začne izvrševati samo testiranje oziroma skeniranje za ranljivostmi (*vulnerability scan*). Če rezultati pokažejo določene bolj kritične ranljivosti, se jih takoj odpravi.

Varnostni inženir informacijskih sistemov, ki vodi aktivnosti in koordinacijo z zunanjim izvajalcem varnostnega testiranja na razpisu izbere podjetje, ki je izvajalec penetracijskega testiranja in ki praviloma to opravlja za dobo nekaj let.

Izvajalec pripravi poročilo o ranljivostih in jih klasificira glede na nivo tveganja; navadno so ti nivoji: visoko tveganje, srednje tveganje, nizko tveganje in informativni nivo. Za vsako od odkrite ranljivosti pripravi opis ranljivosti, scenarij ponovitve in predlog ali priporočilo za ublažitev oziroma odpravo ranljivosti. Poročilo dobi v elektronski obliki, pripravi pa tudi predstavitev rezultatov odgovornim zaposlenim v združbi (varnostni inženirji, vodje oddelkov, razvijalci). Varnostni inženir je dolžan pripraviti poročilo o odzivih na identificirane varnostne ranljivosti, ki jih predstavi vodstvenemu kolegiju združbe. Odzive pripravi v sodelovanju z razvijalci posamezne aplikacije.

Ranljivosti visokega tveganja je združba dolžna v produkcijskem okolju odpraviti v roku enega meseca, vse ostale pa v roku treh mesecev.

4.5 Pridobljena dokumentacija PCI DSS standarda

SUVI standardi prinesejo tudi ustvarjanje in vzdrževanje dokumentacije, ki vsebuje navodila, postopke, politike in ostale vrste dokumentov. Podjetje je tudi zaradi implementacije PCI DSS pridobilo mnogo pomembnih dokumentov, ki zagotavljajo, da so postopki pri udejanjanju varnosti dobro opredeljeni in dokumentirani. Lahko rečemo, da je omenjeno dokumentno gradivo *output* oz. rezultat tudi implementacije PCI DSS.

Več o dokumentaciji in upravljanju z njo bom pisal v naslednjem poglavju v razdelku 5.5.1.

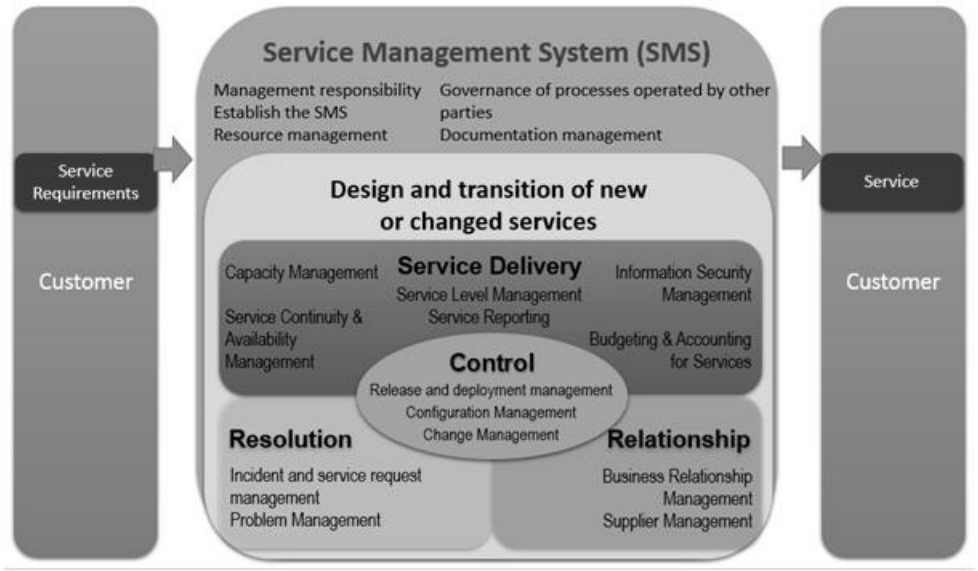
5 Standard ISO/IEC 20000 in ogrodje ITIL

5.1 Predstavitev ISO/IEC 20000 standarda

ISO/IEC 20000[20] standard je mednarodni standard za upravljanje storitev iz področja informacijskih tehnologij (IT storitev). Popolno ime standarda je ISO/IEC 20000 Part 1: Service management system requirements (okrajšano: ISO/IEC 20000-1:2011); je izdelek dveh združb: itSMF and BSI (British Standards Institution). Zadnja verzija, ki je tudi trenutno aktualna, je iz aprila leta 2011.

Kakovost v upravljanju IT storitev je eden od najpomembnejših predpogojev, če želimo, da bodo storitve v poslovnem svetu prepoznane in visoko cenjene. ISO/IEC 20000 standard, v nadaljevanju tudi samo ISO/IEC 20000, podaja minimalen nabor zahtev, ki jih morajo združbe, ki strankam nudijo svoje storitve, izpolniti. Na nek način gre za neodvisno osnovo, iz katerih se lahko storitve potem še izboljšujejo. Podjetje, ki želi pridobiti certifikat ISO/IEC 20000, mora torej izpolniti 256 zahtev; pred sabo dobi seznam procesov, ki jih mora implementirati, delijo se na dva dela:

- ISO/IEC 20000-1 – nabor zahtev, ki morajo biti izpolnjene
- ISO/IEC 20000-2 – podrobnejše smernice, kako zahteve izpolniti



Slika 16 - Shematski prikaz procesov ISO/IEC 20000
Vir: [51]

5.2 Predstavitev ITIL

5.2.1 Opredelitev ITIL

ITIL je okrajšava za *Information Technology Infrastructure Library*, to pa bi lahko prevedli kot knjižnica za infrastrukturo informacijske tehnologije.

V naslednjih dveh alinejah je podana opredelitev ITIL [52]:

- Zagotavlja celovit, dosleden in skladen nabor najboljših praks za upravljanje procesa storitev s področja informacijske tehnologije

[20] ISO standardi so v lasti združbe ISO, ki je bila ustanovljena 23. 2. 1947 v Ženevi, Švica. Združba razglaša svetovne industrijske in tržne standarde. Ima skupno 162 [50] držav članic (2015).

- Spodbuja kakovosten pristop k doseganju poslovne učinkovitosti in učinkovitosti pri uporabi informacijskih sistemov

Jedro standarda ITIL, ki je bil razvit pri *UK Office of Government Commerce*[21] z namenom priprave standarda in dokumentacije standardiziranega nabora poslovnih praks za IT storitve, sestavljata dve široki skupini procesov, in sicer nudenje storitev in podpora tem storitvam [26]. Storitev je torej glavna »sestavina« v standardu in je po opredelitvi [5] način zagotavljanja dodane vrednosti uporabnikom tako, da uresničujejo tisto, kar uporabniki želijo, ne da bi pri tem nase prevzemali določene stroške ali tveganja. Upravljanje storitev je množica specializiranih organizacijskih zmožnosti, s katerimi v obliki storitev prinašamo dodano vrednost.

ITIL je integriran nabor najboljših praks in procesov za zagotavljanje IT storitev, ki jih združba ponuja strankam. Glavni poudarek je maksimiranje (poslovnih) vrednosti za kupce z uskladitvijo IT sredstev s poslovnimi potrebami.

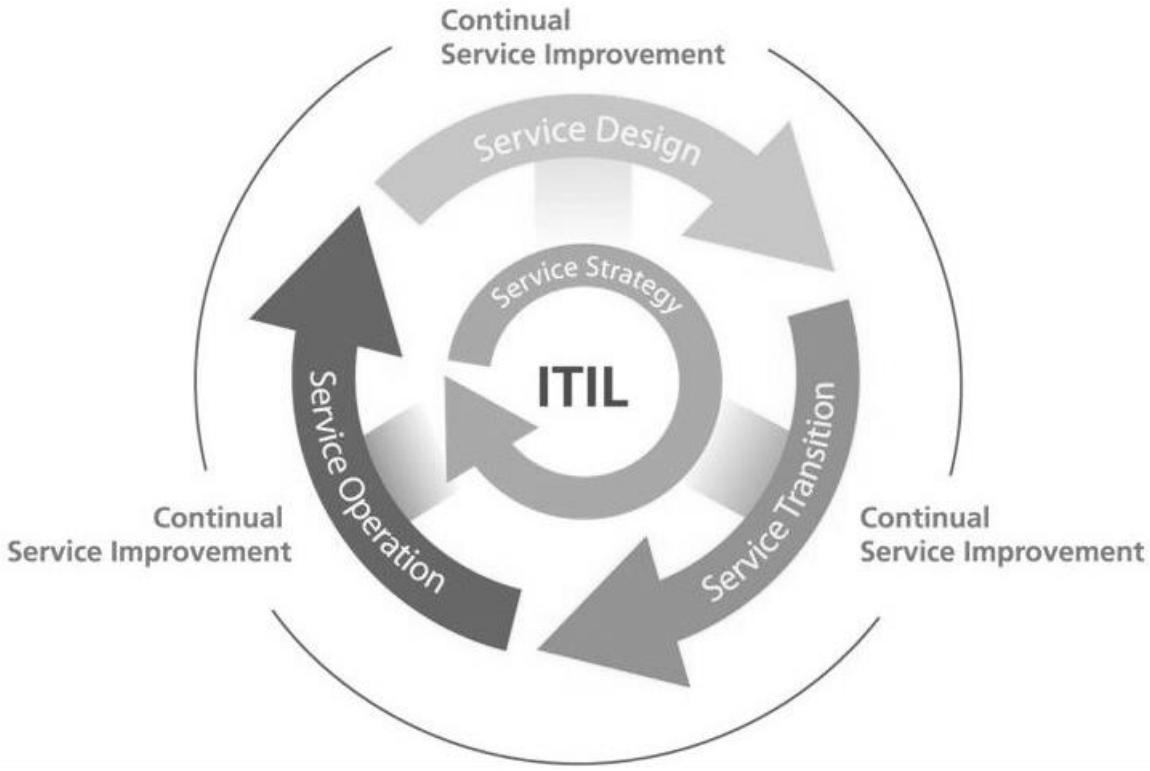
ITIL vsebuje podrobnejše opise procesov, tokov, dejavnikov uspeha, metrik in implementacijskih napotkov, ki jih združbe priredijo na način, da le-te v njihovem okolju delujejo.

Ogrodje ITIL lahko pomaga združbam oblikovati in izboljšati sposobnost upravljanja njihovih IT storitev, povečati uskladitev in maksimirati poslovanje in poslovanju dokazati vrednost.

ITIL ni namenjen temu, da bi natančno povedal, na kakšen način in v kolikšni meri procese ogrodja prilagoditi združbi in ali je sploh potrebno določeno potrebo prepoznati in jo prilagoditi. Vsak proces ima namreč dokumentirano vrednost za poslovanje in je lahko uporabljen in prilagojen posebej. Vendarle je potrebno poudariti, da je večina procesov v določeni soodvisnosti do drugih in je na ta način procese težko prilagoditi popolnoma neodvisno od drugih.

ITIL je de facto standard za implementacijo ITSM [53], kjer je ITSM ali *Information Technology Service Management* oziroma Upravljanje IT storitev strateški pristop k načrtovanju, ponujanju, upravljanju in izboljševanju uporabe informacijske tehnologije v združbi. Cilj ITSM je zagotoviti prave procese, človeške vire in tehnologijo na pravem mestu in s tem pripomoči združbi doseči poslovne cilje [54].

[21] UK Office of Government Commerce (OGC) je bila agencija vlade Združenega kraljestva Velike Britanije in Severne Irske)



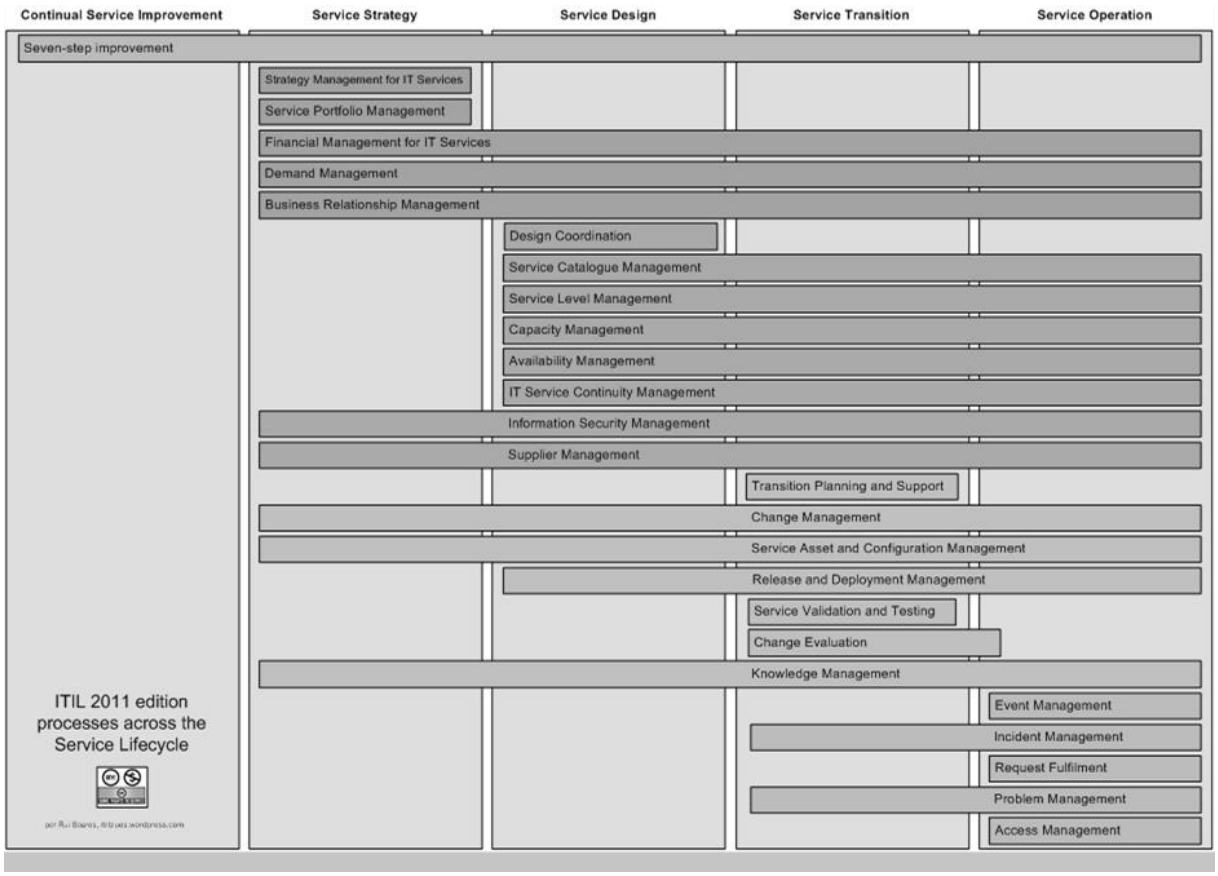
Slika 17 - ITIL, življenjski cikel storitve

Strategija -> Načrtovanje –> Prenos –> Delovanje -> Izboljševanje
Vir: [55]

Glede na sliko 17, ki prikazuje življenjski cikel IT storitve, lahko po standardu ITIL identificiramo 5 osnovnih področij oz. faz storitev:

1. Strategija storitev: razumeti organizacijske cilje in potrebe uporabnikov IT storitev
2. Načrtovanje storitev: oblikovati strategijo storitev v načrt pridobivanja poslovnih ciljev
3. Prenos storitev: razviti in izboljšati zmožnosti za uvedbo novih storitev v podprta okolja
4. Delovanje storitev: upravljanje storitev v podprtih okoljih
5. Stalno izboljševanje storitev: storitve stalno izboljševati

Zgoraj opisana področja v verziji 3 standarda ITIL združujejo 26 procesov. Slika 18 prikazuje grupiranje procesov znotraj področij. Videti je mogoče, da se večina procesov razteza čez več področij, vendarle ima vsak svoje matično področje.



Slika 18 - Procesi, združeni v področja
Vir: [56]

5.2.2 Področja/faze ITIL

V nadaljevanju na kratko opisal vsako od v prejšnjem razdelku omenjenih petih področij življenjskega cikla storitev in naštel njihove procese.

5.2.2.1 *Strategija storitev*

Cilj področja strategije storitev je zagotoviti združbam sposobnost za oblikovanje, razvoj in izvajanje storitev vodenja kot strateško prednost in razmišljati in delovati na strateški način. Zajema naslednje procese:

- Upravljanje IT storitev
- Upravljanje portfelja storitev
- Finančno upravljanje storitev
- Upravljanje zahtev
- Upravljanje poslovnih razmerij

5.2.2.2 *Načrtovanje storitev*

Glavni cilj te faze je načrtovanje novih ali spremenjenih storitev s ciljem postavitve v produkcijsko okolje. Zajema naslednje procese:

- Koordinacija načrtovanja
- Upravljanje kataloga storitev
- Upravljanje na nivoju storitev
- Upravljanje razpoložljivosti
- Upravljanje zmožnosti in kapacitet

- Upravljanje kontinuitete IT storitev
- Upravljalni sistem za informacijsko varnost
- Upravljanje dobaviteljev

5.2.2.3 *Prenos storitev*

Faza prenosa storitve omogoča, kot že samo ime pove, prenos storitev v produkcijsko oz. obratovalno okolje. Procesi so naslednji:

- Načrt prenosa in podpora
- Upravljanje sprememb
- Upravljanje sredstev in konfiguracij
- Upravljanje izdaj in namestitev
- Validacije storitev in testiranje
- Evaluacija sprememb
- Upravljanje znanja

5.2.2.4 *Delovanje storitev*

Faza delovanja storitev je odgovorna za tekoče upravljanje tehnologije, ki je uporabljena za glavne in podporne storitve z naslednjimi procesi:

- Upravljanje dogodkov
- Upravljanje incidentov
- Izpolnjevanje zahtev
- Upravljanje problemov
- Upravljanje identitet

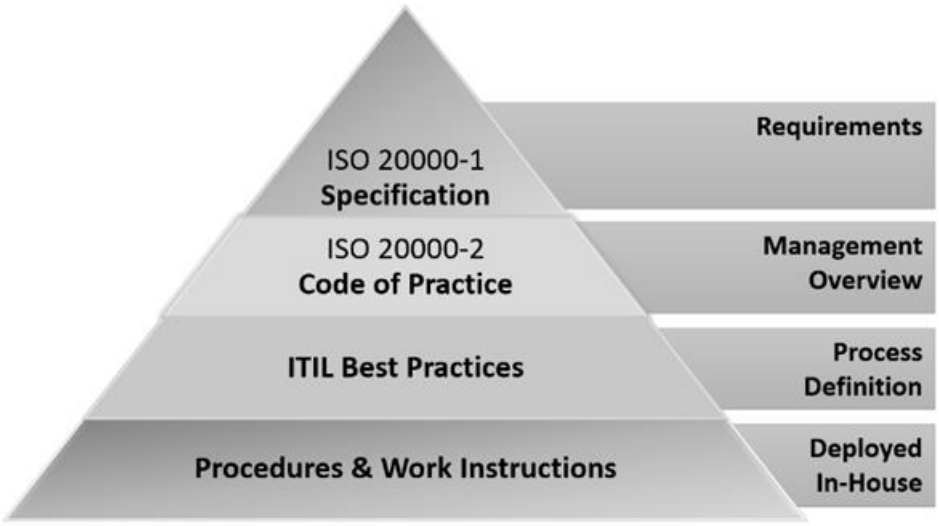
5.2.2.5 *Stalno izboljševanje storitev*

Namen te faze je identificirati spreminjajoče se poslovne potrebe, jih uskladiti in preusmeriti. S tem zagotovimo tudi stalno izboljševanje storitev IT, ki podpiraje poslovne procese. V tej fazi izboljšujemo tudi kvaliteto storitev – izboljšujemo kakovost, učinkovitost in stroškovno učinkovitost IT procesov skozi celoten življenjski cikel ob pogoju, da je točno opredeljeno, kaj naj bo nadzorovano in evaluirano. Faze v tem področju so:

- Merjenje storitev
- Poročanje o storitvah
- Izboljševanje storitev

5.3 Povezava med ogrodjem ITIL in ISO/IEC 20000 standardom

Eden glavnih razlogov za novo verzijo ITIL V3 je bil ravno doseči boljšo povezavo z ISO/IEC 20000 standardom. Diagram spodaj (Slika 19) dobro prikazuje relacijo med njima. Na vrhu so zahteve ISO/IEC 20000, ki morajo biti izpolnjene, nižje so smernice in priporočila, kako te zahteve izpolniti. Temu dvojemu so osnova najboljše prakse ITIL. Čisto na dnu piramide so postopki in navodila osnovnih dejavnosti samega podjetja, pa tudi prakse ostalih implementiranih standardov.



Slika 19 - Piramidni prikaz povezave med standardom ISO/IEC 20000 in ogrodjem ITIL
Vir: [51]

Z izpolnitvijo najboljših praks ITIL lahko torej avtomatsko dosežemo zahteve ISO/IEC 20000 ob pogoju, da so le-te izpolnjene kakovostno in dosledno.

Ob koncu morda še to: zanimivo je, da certifikacija ITIL poteka individualno, torej za posameznike; tudi v podjetju, obravanavanem v tem magistrskem delu, se skrbniki poslovnih sistemov izobražujejo in poskušajo pridobiti ITIL certifikate (obstajajo začetni, nadaljevalni in ekspertni nivoji). Po drugi strani pa se ISO/IEC 20000 osredotoča na združbe, certifikat pridobi podjetje samo in ne posamezniki znotraj njega. Tako le-ta postane intelektualna lastnina podjetja.

5.4 Podrobnejši opis področja Prenos storitev in procesa Upravljanje sprememb

5.4.1 Uvod

V razdelku 3.4.6.1 sem pod točko T3.1 identificiral tveganje v obliki nameščanja nepravilne verzije aplikacije v produkcijsko okolje. V naslednjih dveh razdelkih bom najprej podorbneje predstavil področje Prenos storitev ogrodja ITIL (5.4), nato pa še na primeru omenjenega identificiranega tveganja v praksi predstavil izvedbo v združbi s področja Prenosa storitev (5.4).

5.4.2 Splošno o področju Prenos storitev

To področje povezuje vse, kar je potrebno, da se posamezna poslovna storitev prenese v živo in produkcijsko uporabo [57]. Dodane vrednosti, ki naj bi jih procesi iz tega področja doprinesli poslovanju podjetja, so med drugim omogočiti spremembe poslovanja, zmanjšati vpliv na poslovanje, ki bi sicer lahko vplivali, če bi bile spremembe upravljane na nepravilen ali neurejen način in pridobiti prednost pri poslovanju z novimi storitvami. Prav tako to področje zagotavlja, da so načrt in koncepti storitev implementirani, kot se pričakuje in zagotavlja, da je sistem za upravljanje s storitvami pripravljen za delo z novimi in spremenjenimi storitvami. Na koncu, pa ne nujno najmanj pomembno pa je, da zmanjšuje število napak, ki bi se lahko zgodile v produkcijskem okolju.

Cilji, ki naj bi jih po ITIL verzije 3 pri področju prenosa storitev zasledovali, so [52]:

- Pripraviti pričakovanja strank o tem, kako naj bo storitev uporabljana s katerim omogočimo poslovne procese
- Koordinirati izdaje novih storitev med poslovnimi in tehničnimi spremembami oz. projekti
- Zmanjšati morebitne razlike med pričakovanim in dejanskim obnašanjem prenesene storitve

- Zmanjšati znane napake in tveganja v zvezi s prenosom storitve
- Zagotoviti, da storitev izpolnjuje podane zahteve

Najpomembnejša vloga področja prenosa storitev je upravljalec prenosa storitev; zadolžen je, da vsakodnevno upravlja in nadzoruje aktivnosti prenosa storitev, ki jo opravljajo zadolžene skupine.

Kot sem že pisal v razdelku 5.2.2.35.2.2.3, je področje prenosa storitev sestavljeno iz naslednjih procesov:

- Načrt prenosa in podpora
- Upravljanje sprememb (*Change Management*)
- Upravljanje sredstev in konfiguracij (*Service Asset and Configuration Management*)
- Upravljanje izdaj in namestitev (*Release and Deployment Management*)
- Validacije storitev in testiranje
- Evalvacija sprememb
- Upravljanje znanja

ITIL predvideva sistem oz. bazo znanja o storitvah (*Service Knowledge Management System* (SKMS)); to je skupek orodij in ene ali več podatkovnih baz, v katerih je zabeleženo vse potrebno, da lahko upravljamo s celim življenjskim ciklom posamezne storitve. SKMS vsebuje konfiguracijski upravljavski sistem (*Configuration Management System* (CMS)).

5.4.3 Proces Upravljanje sprememb (*Change Management*)

5.4.3.1 *Opredelitev*

Ta proces je eden ključnih in najpomembnejših znotraj te faze življenjskega cikla storitve, zagotavlja pa, da so spremembe zapisane in nato evaluirane, avtorizirane (torej imajo svoje nosilce in odločevalce), urejene po prioritetah, načrtovane, testirane, implementirane, dokumentirane in pregledane na kontroliran in urejen način.

Kot je videti iz diagrama »Procesi, združeni v področja« iz poglavja 2, je upravljanje storitev proces, čigar udeleženost se razteza skozi ves življenjski krog storitve.

5.4.3.2 *Terminologija in pojmi*

V nadaljevanju bom navedel in opisal nekaj ključnih pojmov za razumevanje in implementacijo procesa upravljanja sprememb.

Sprememba (*Change*)

Kratko in jedrnato je po ITIL verzije 3 sprememba dodajanje, spreminjanje ali odstranjevanje vsega, kar ima lahko vpliv na IT storitve.

ITIL predvideva naslednje koncepte oz. vrste sprememb:

- Normalna sprememba: dovolj velika in pomembna sprememba, ki zahteva zahtevek za spremembo (RFC) – glej naslednji razdelek.
- Standardna sprememba: pred-odobrena sprememba, ki ima nizko stopnjo tveganja. Sprememba je lahko narejena v skladu z vnaprej pripravljenimi navodili; kot primer lahko navedemo posodobitev šifrantov v produkcijski bazi.
- Nujna sprememba: tovrstna sprememba mora biti opravljena karseda hitro; navadno gre za napake v delovanju sistema, ki povzročajo poslovno škodo ali pa nujni varnostni popravki.

Zahtevek za spremembo (*Request For Change (RFC)*)

Zahteva za spremembo je formalni zahtevek za spremembo neke zaključene funkcionalne celote na nekem produktu ali sistemu.

Pri projektnem upravljanju zahtevek nastane takrat, ko stranka želi dodatno funkcionalnost ali spremembo obstoječe funkcionalnosti. Takšna sprememba poleg ostalega lahko vsebuje npr. dodatno funkcijo, razširitev funkcionalnosti ali izboljšavo delovanja. Do zahtevkov za spremembo lahko pride iz notranjih potreb podjetja, še večkrat pa so naročniki sprememb zunanje stranke, ki večinoma za spremembe tudi plačajo.

Pri slednjem je najbolj pomembno to, da so vse podrobnosti natančno opredeljene in da obe strani natančno razumeta, kaj pričakovati od zahtevka za spremembo. Seveda je potrebna jasna in natančna dokumentacija.

Kot že rečeno, so drugi izvor sprememb notranje potrebe podjetja. To so največkrat akcije v obliki nameščanja varnostnih in drugih popravkov na strojni in programski opremi ali spremembe pri programski opremi, ki jih podjetje interno uporablja.

Ko je zahtevek za spremembo implementiran, je nujen pregled implementacije tako na tehničnem kot na vsebinskem nivoju, s katerim preverimo, ali je bila sprememba uspešna in učinkovita.

Vloge v procesu sprememb (*Change Process Roles*)

Kakor ostali procesi v okviru ITIL, ima tudi proces upravljanja sprememb nekaj ključnih vlog. Naj jih naštejem nekaj:

- Upravitelj sprememb (*Change Manager*): Upravitelj sprememb je krovna oseba/vloga, ki skrbi za nadzor nad načinom udejanjanja procesa upravljanja sprememb. Skrbi, da aktivnosti v zvezi s procesom potekajo nemoteno v skladu s pravili in zadolžitvami, da so ostale vloge porazdeljene.
- Posvetovalni tim za spremembo (*Change Advisory Board (CAB)*): Ekipa ljudi, navadno vodje posameznih oddelkov in odločevalci, ki se seznanijo z načrtovano spremembo in jo potrjujejo oz. zavrnejo, če se ugotovi, da zahtevek ne zadošča vsem predpisanim kriterijem in okoliščinam. Te okoliščine, v katerih bo sprememba narejena, so med drugim tudi: termin implementacije spremembe, obveščanje zunanjih partnerjev o spremembah in prekinitvah pri implementaciji, seznam sodelujočih pri spremembi itd.
- Posvetovalni tim za nujno spremembo (*Emergency Change Advisory Boards (ECAB)*): Ekipa ljudi, prav tako navadno iz vodstvenega in odločevalskega kadra, ki imajo pravico posamič in izven reda odobriti nujne spremembe zaradi izpada ali napak v produkciji.

Urnik sprememb (*Change Schedule*)

Urnik sprememb drži seznam vseh potrjenih predlogov za spremembe in njihovih načrtovanih datumov implementacije.

Predviden izpad storitve (*Projected Service Outage - PSO*)

Predviden izpad storitve je pojem, ki drži seznam vseh pričakovanih razlik nad razpoložljivostjo storitev kakor je sporazumljeno v SLA-jih. SLA je na kratko okrajšava za *Service-level agreement*, pogodbo, ki določa formalno opredelitev storitev.

5.5 Prenos storitev – primer v podjetju

5.5.1 Uvod – splošno o implementaciji ITIL v podjetju

Podjetje je tudi zaradi implementacije ITIL pridobilo mnogo pomembnih dokumentov, ki zagotavljajo, da podjetje posluje s cilji, ki jih zadaja ITIL in standardi, ki jih narekuje standard ISO/IEC 20000. Lahko rečemo, da je omenjeno dokumentno gradivo *output* oz. rezultat tudi implementacije ITIL. Podjetje je tako, kot rečeno, tudi zaradi ITIL pridobilo krovni dokument strategija in cilji, ki jih najpomembnejši dokument podjetja in je v njem zapisano tisto, k čemur naj bi združba stremela.

Velja poudariti, da si le-ti sledijo po pomembnosti od najpomembnejšega do (morda) manj pomembnih; lahko pogledamo tudi skozi prizmo abstrakcije in agregacije vsebine: strategija je dokument, ki v celoti in najbolj abstraktno zajema vizijo poslovanja združbe, po drugi strani pa je vsebina operativnih navodil, zapisov, zunanjih in tehničnih dokumentov dejanska izvedba nalog itd. iz (skoraj) vseh področij poslovanja združbe. Seznam tipov dokumentov je torej naslednji:

- strategija
- politika
- pravilnik
- operativno navodilo
- zapis
- zunanji dokumenti
- tehnični dokumenti

Vsak od dokumentov ima lastnika dokumenta, ki je odgovoren za vsebino in ki opravi vsakoletni pregled dokumenta, s katerim se potrjuje njegova veljavnost.

Dokumenti se hranijo v združbi razvitem dokumentnem sistemu v orodju Sharepoint, ki omogoča tudi skupinsko delo na dokumentih in odobravanje dokumentov s strani nosilca/lastnika ter redne periodične in izredne preglede.

5.5.2 Implementacija procesa Prenos storitev - Upravljanje sprememb v praksi

V tem podpoglavju bom podrobneje opisal področje prenosa storitev v podjetju z največjim poudarkom na procesu upravljanja sprememb; od vseh področij ITIL ravno pri tem tudi sam v največji meri sodelujem pri implementaciji tega procesa.

Kot sem pisal v poglavju 5.4.1, je ključen pojem upravljanja sprememb – sprememba. Opredelil in opisal sem tudi glavne pojme upravljanja sprememb, kot so zahtevek za spremembo, kategorije sprememb, vloge pri spremembah, urniki sprememb in predviden izpad storitev zaradi nadgradenj. Vse te ključne pojme bom v spodnjih razdelkih vključil v opis tranzicije storitev v produkcijsko okolje v združbi.

5.5.2.1 *Primer sistema, storitve*

Sam delam kot razvijalec (tudi) na področju spletnih storitev. Strankam nudimo uporabo spletnih storitev s katerimi le-te beležijo aktivnosti v sisteme. Za to beleženje uporabljajo nabor metod spletnih storitev, opis metod, vhodne parametre in izhodne parametre pa opredeljuje WSDL (*web service definition language*). Vsako pomlad in jesen je predvidena izdaja novega WSDL z novimi funkcionalnostmi in spremembami WSDL. Naj omenim, da so spletne storitve in s tem WSDL le del spletnega sistema, ki vključuje tudi spletne aplikacije, registracijske sisteme in infrastrukturo za komunikacijo z zunanjimi partnerji.

Zahtevki za spremembe v večji meri prihajajo od strank – zunanjih partnerjev, ki želijo izboljšave in dodatne funkcionalnosti storitev; v manjši meri pa tudi iz notranjih zahtev podjetja.

V opisanem primeru spletnih storitev tako zahtevki za spremembo (vsebinsko) vedno pridejo s strani strank – npr. nova metoda v WSDL z novo funkcionalnostjo.

V razdelku 3.2.2 – Terminologija in pojmi sem opisal najpomembnejše pojme, ki se pojavljajo v zvezi s upravljanjem sprememb; v spodnjih vrsticah navajam konkreten primer:

- Sprememba: Dodatna metoda v WSDL
- Zahtevek za spremembo: Začet s strani stranke, odprt s strani tehnologije
- Vloge v procesu sprememb
    - Upravitelj spremembe (Change Manager): Vodja razvojnega oddelka

    - Posvetovalni tim za spremembo (Change Advisory Board (CAB)): Vodje oddelkov, direktorji
    - Posvetovalni tim za nujno spremembo (Emergency Change Advisory Boards (ECAB)): /
- Urnik sprememb: 15. november 2014 ob 22:00 do 24:00
- Predviden izpad storitve: 10 minut med 22:00 in 24:00

Še primeri vrst sprememb:

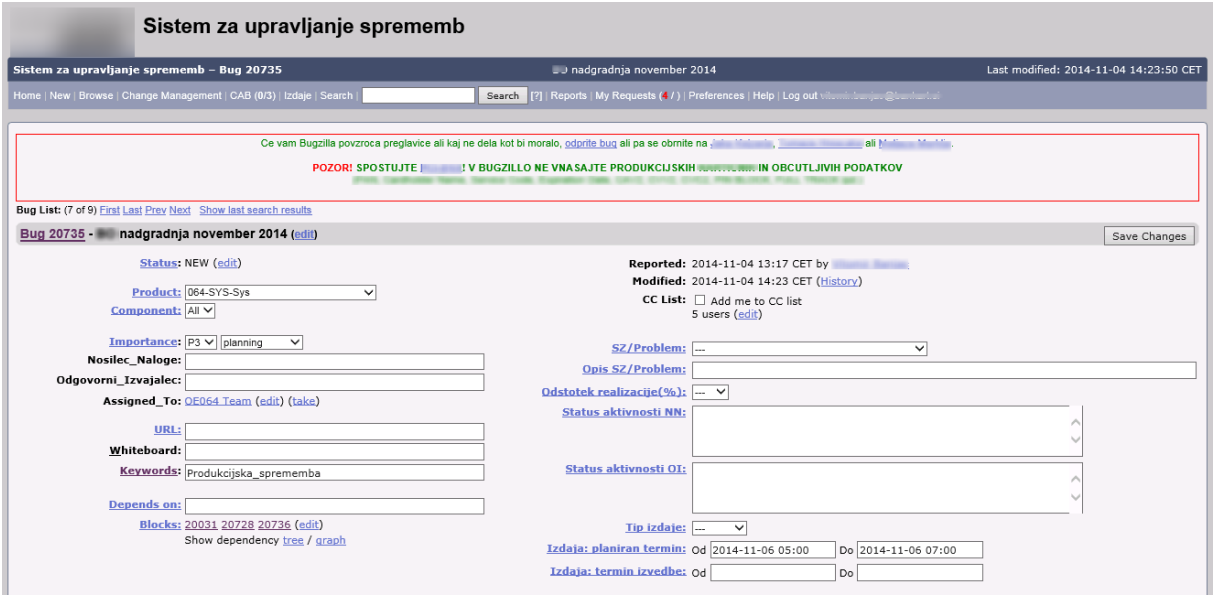
- Vrste sprememb
    - Običajna sprememba: Dodatna metoda v WSDL
    - Standardna sprememba: Dodajanje zapisov v šifrant v podatkovno bazo, ki ga uporabljajo spletne storitve

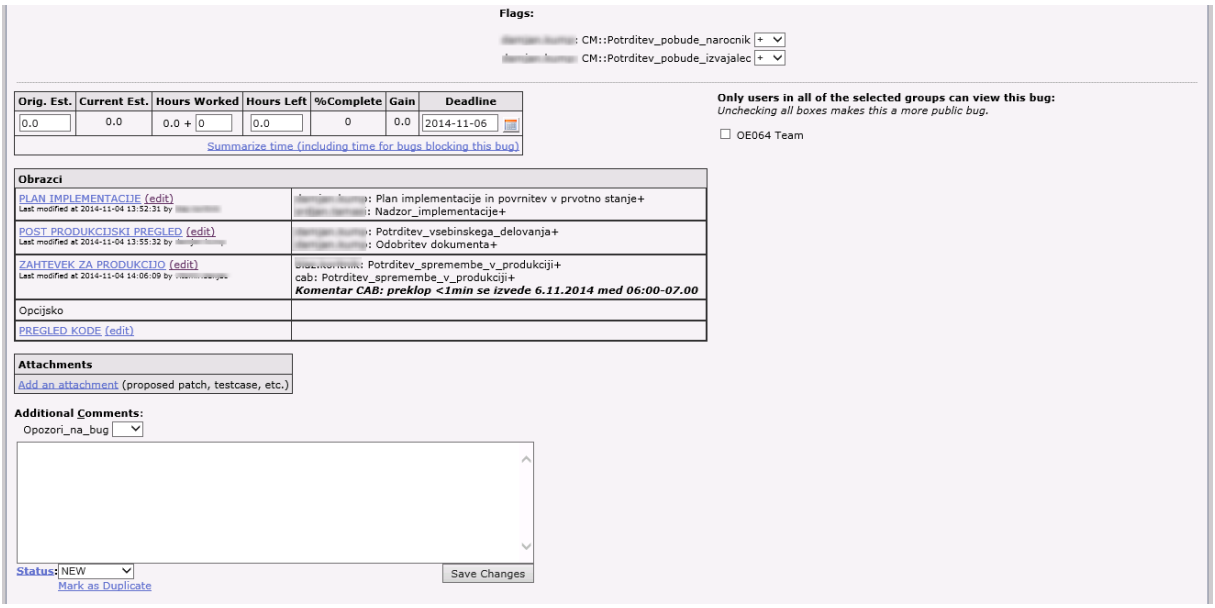
Nujna sprememba: Nujna odprava napake zaradi nepravilnega delovanja metode – metoda vrača preveč zapisov

5.5.2.2 *Uporaba Bugzille kot sistema za upravljanje sprememb*

Bugzilla je odprtokodna spletna programska oprema, ki je v prvi vrsti namenjen sledenju reševanja problemov in napak v programski in strojni opremi (*issue-tracking system*). Napisana je v programskem jeziku Perl.

Zgodovinsko gledano je Bugzilla, tako kot je njen osnovni namen, v podjetju služila kot orodje za sledenje in odpravo zahtevkov in napak v informacijski infrastrukturi in aplikacijah. Kasneje, skozi faze vpeljave izboljšav in novih postopkov, ki jih določata ogrodje ITIL in standard ISO/IEC 20000, je bila v Bugzillo dodana cela vrsta prilagoditev. Tako je v njej možno dodajanje dokumentov, vnašanje ključnih besed, pisanje datumov rokov določenih aktivnosti, potrjevanje aktivnosti z zastavicami in drugo potrebno za podprtje dela po prej omenjenih standardih.





Slika 20 - Zaslonska maska Bugzille kot sistema za upravljanje sprememb.

5.5.2.3 *Aktivnosti ob produkcijski spremembi*

CMDB in verzioniranje

ITIL za sledenje verzij priporoča označevanje posameznih izdaj. Pred prehodom v produkcijo mora tako biti zaradi lažjega upravljanja in kontrole vsaka izdaja programske opreme (npr. WSDL kot primer) označena z enolično označbo. Izdaje se označujejo glede na pomembnost in obseg tako v vsebinskem kot v tehničnem smislu. Tako se največje in najpomembnejše izdaje, ki so najredkejše, označujejo oz. se dvignejo z *major* številko, malo manj pomembne z *minor* številko, manjše popravke z *build* številko in nujne popravke z *revision* oz., s tako imenovano *patch* verzijo.

Ker razvoj poteka v C# programski kodi, se drži nomenklatura <*major version*>.<*minor version*>.<*build number*>.<*revision*>, kot je razvidno tudi spodaj, iz programske kode.

```
// Version information for an assembly consists of the following four values:
//
// Major Version
// Minor Version
// Build Number
// Revision
//
/ You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [assembly: AssemblyVersion("1.0.*")]
```

Slika 21 - Primer C# kode za nastavitev verzije aplikacije

DEV Integration PO	Success	2014-11-12 14:10:05	2014-11-17 21:12:36	2.7.4.0	Running	Sleeping	• Failing Tasks : Svn: CheckForModifications	Force Stop
DEV Integration PO	Failure	2014-11-12 14:10:20	2014-11-17 21:12:36	3.6.17.0	Running	Sleeping	• Breakers : • Failing Tasks : MsBuildTask	Force Stop
DEV Integration	Success	2014-11-12 14:10:19	2014-11-17 21:12:36	2.2.0.0	Running	Sleeping	• Failing Tasks : Svn: CheckForModifications	Force Stop
GUI Publish A	Success	2013-05-26 21:38:25	Force Build Only		Running	Sleeping		Force Stop
GUI Publish A	Success	2011-06-27 12:56:54	Force Build Only	3.6.1.0	Running	Sleeping		Force Stop
GUI Publish B	Failure	2013-10-24 17:02:06	Force Build Only		Running	Sleeping		Force Stop
GUI Publish C	Success	2011-06-27 13:06:50	Force Build Only	2.5.1.0	Running	Sleeping		Force Stop
GUI Publish P	Success	2011-06-27 13:27:35	Force Build Only	2.5.1.0	Running	Sleeping		Force Stop
GUI Publish PC	Success	2011-06-24 14:32:51	Force Build Only	3.5.2.0	Running	Sleeping		Force Stop
RELEASE BUILD At	Success	2014-09-23 14:17:48	Force Build Only	1.7.7.0	Running	Sleeping		Force Stop
RELEASE BUILD A	Success	2014-11-12 10:14:34	Force Build Only	3.7.3.0	Running	Sleeping		Force Stop
RELEASE BUILD Lib	Success	2014-11-17 13:34:03	Force Build Only	2.2.0.3	Running	Sleeping		Force Stop

Slika 22 - Zaslonska maska spletne aplikacije strežnika za pripravo izdaj (build server)

Pregled kode (*Code Review*)

Pregled kode je ena od aktivnosti, ki se izvede pred produkcijsko spremembo in doprinese pomemben delež h kvaliteti implementacije in zmanjšanju napak. Pregled kode opravi sodelavec, ki ni neposredno sodeloval pri implementaciji sprememb, pozna pa materijo in vsebino sistema, na katerem izvaja pregled. Rezultat pregleda kode je dokument Pregled kode.

Strežnik za pripravo namestitvenih verzij (*Build Server*)

Za pripravo izdaj programske opreme z novimi zahtevami in/ali popravki, se v podjetju uporablja strežnik za pripravo verzij (*build server*), implementiran v CruiseControl.NET (CC.net) ogrodju. Izvorno je CruiseControl javanski odprtokodni projekt za neprekinjeno avtomatsko izvajanje izdaj, v podjetju uporabljana različica pa je napisana v programskem jeziku .NET.

Za produkte/projekte se uporablja dve vrsti priprave izdaj (*buildov*): integracijske in produkcijske. Sistem je nastavljen tako, da integracijske *builde* izvaja po vsakem uspešnem dodajanju sprememb (*commitu*) v verzijski sistem (*source control*) Subversion. Če *build* ne uspe, se pošlje elektronska pošta osebi, ki je oddala ne-delujočo kodo, z informacijo o napaki; le-ta je obvezan, da kodo ustrezno popravi in ponovno odda. S tem se zagotavlja kontinuiteta kvalitete kode, ki se lahko uspešno *builda*. Bolj pomembni od integracijskih *buildov* so prej omenjeni produkcijski *buildi*, ki se izvedejo na zahtevo. Izdelek so binarne datoteke, ki se jih namesti ob produkcijski spremembi. Vsaka produkcijska izdaja (*build*), ki se naredi, je enolično označena z *assembly* verzijo.

```
    <Exec Command="xcopy $(CommonLibFolder)\[illegible] $(SolutionFolderLibDir)\ /e /Y" />
    <Exec Command="xcopy $(CommonLibFolder)\[illegible] $(SolutionFolderLibDir)\ /e /Y" />
    <Exec Command="xcopy $(CommonSharedFolder)\[illegible] $(SolutionFolderLibDir)\ /e /Y" />
</Target>

<ItemGroup>
    <ToCommitLibs Include="$(SolutionFolderLibDir)" />
</ItemGroup>

<Target Name="CommitReleaseLibs">
    <Message Text="Commiting changes to SVN"/>
    <SvnCommit Message="Release lib - $(CCNetLabel) $(TimeStamp)"
           Targets="@(ToCommitLibs)">
    </SvnCommit>
</Target>
<!-- END Copy LIB files, commit -->

<!-- Get the projects from the solution -->
<Target Name="GetProjectsFromSolution">
    <GetSolutionProjects Solution="$(SolutionFolderFullPath)\[illegible].sln">
        <Output ItemName="ProjectFiles" TaskParameter="Output"/>
    </GetSolutionProjects>
</Target>

<!-- Compile each of the selected projects -->
<Target Name="CompileProject" DependsOnTargets="GetProjectsFromSolution">
    <Message Text="Building the project Test-$(ReleaseLabel)" />
    <MSBuild
        Projects="$(SolutionFolderFullPath)\%(Identity)"
        Properties="Platform=x86;
        Configuration=Release;
        OutDir=$(FullReleaseDirPath)\%(ProjectFiles.Filename)\;
        TrackFileAccess=false"
        Targets="Clean;Rebuild">
    </MSBuild>
</Target>

<Target Name="CopyToProducts">
```

Slika 23 - Primer skripte za pripravo produkcijskih namestitvenih verzij

5.5.2.4 *Priprava dokumentacije pred implementacijo sprememb*

V nadaljevanju bom opisal zahtevke pred implementacijo sprememb, kot se vodijo v podjetju.

Zahtevek za spremembo

Zgoraj opisani primer spremembe WSDL v Bugzilli je voden kot en (razvojni) zahtevek za spremembo pod enim hroščem (*bugom*), ki ga odpre tehnolog – vsebinski skrbnik spletnih storitev. Ta zahtevek za spremembo je osnovni dokument za implementatorja – razvijalca, ki naredi spremembe spletnih storitev v skladu s specifikacijo, ki je navadno objavljena v zahtevku. Pred začetkom razvoja se v hrošču opredeli tudi velikost izdaje, *deadline* oz. rok izdaje, odgovoren nosilec naloge, zastavica za potrditev pobude nadrejenemu itd.

Ko je razvoj zaključen, se pripravi vsa potrebna dokumentacija, da se sprememba potrdi s strani vodstva na CAB sestanku. Vsa dokumentacija, vezana na zahtevek za spremembo, se doda na pripadajoč razvojni hrošč oz. zahtevek. Dokumentacija vsebuje naslednje dokumente:

- Analiza vpliva: dokument, v katerem naročnik spremembe (s poslovnega/vsebinskega vidika) in razvijalec (s tehničnega vidika) opišeta vpliv spremembe. Tehnični del vsebuje seznam kodnih sprememb in seznam sprememb z vidika varnosti; vsebinski pa opiše vpliv sprememb na delovanje sistema.
- Testiranje: dokument, v katerem naročnik spremembe vpiše scenarij testiranja oz. testni plan in po končanem dejanskem testiranju vpiše rezultate. Testi se izvajajo v razvojnem okolju. Ti morajo biti uspešni, če želimo, da je zahtevek oddan v odločanje. Razvijalec je po drugi strani prav tako dolžan pripraviti testni scenarij, na razvojnem okolju izvesti testiranje s poudarkom

na tehničnem delu in rezultate vpisati v dokument. Tudi rezultati testiranja tehničnega dela morajo seveda biti pozitivni.
- Pregled kode: opcijski dokument; tehnični skrbniki lahko pripravijo seznam kodnih sprememb, ki jih nato pregleda drug razvijalec, ki ni neposredno implementiral spremembe. Le-ta poda svoje mnenje in pripombe o razviti kodi ali ostalih tehničnih podrobnostih.

Simulacijska sprememba

Predpriprava na produkcijsko spremembo je simulacijska sprememba, ki se jo opravi na simulacijskem okolju. Simulacija je okolje, ki v vseh komponentah, tako v strojni in komunikacijski opremi kot tudi v nameščenih komponentah, programski opremi, konfiguracijah in ostalem zasleduje produkcijo – tako je na voljo okolje, ki je zadnji »test« pred prehodom na produkcijo.

Običajno se simulacijska sprememba izvrši nekaj dni pred načrtovano produkcijsko spremembo s ciljem, da je na razpolago dovolj časa, da tehnologi in/ali stranke preizkusijo implementirane spremembe v tem okolju. Za naš primer bi to pomenilo, da novo verzijo spletnih storitev namestimo na simulacijsko okolje in je tako nov WSDL na voljo strankam za testiranje.

Tudi za simulacijsko spremembo je v Bugzilli potrebno na ustrezen način odpreti hrošč. V njem se vodi simulacijska sprememba. Pripraviti je potrebno naslednje dokumente:

- Analiza vpliva: podobno kot v zahtevku za spremembo je to dokument, v katerem naročnik spremembe (s poslovnega/vsebinskega vidika) in razvijalec (s tehničnega vidika) opišeta vpliv spremembe v simulacijskem okolju. Tehnični del vsebuje seznam kodnih sprememb in seznam sprememb z vidika varnosti (v simulaciji).
- Testiranje: spet podobno kot v zahtevku za spremembo - dokument, v katerem naročnik spremembe vpiše scenarij testiranja oz. testni plan in po končanem dejanskem testiranju vpiše rezultate. Testi se izvajajo v testnem okolju. Ti morajo biti uspešni, če želimo, da je zahtevek oddan v odločanje. Razvijalec je po drugi strani prav tako dolžan pripraviti testni scenarij, na testnem okolju izvesti testiranje s poudarkom na tehničnem delu in rezultate vpisati v dokument. Tudi rezultati testiranja tehničnega dela morajo seveda biti pozitivni.
- Plan implementacije in povrnitev v prvotno stanje: v ta dokument je razvijalec dolžan vpisati podrobnejši plan poteka spremembe v simulaciji: uro začetka izvajanja spremembe, natančen popis aktivnosti implementacije simulacijske spremembe; pravzaprav navodila kako izvesti aktivnosti. Pomemben del dokumenta je tudi rubrika »Povrnitev v prvotno stanje«, kamor izvajalec vpiše postopek v primeru težav pri prehodu v simulacijo ali pa ob težavah/nepravilnem delovanju implementirane spremembe. Potrditev dokumenta je potrebna s strani drugega razvijalca in s strani nadrejenega.
- Zahtevek za simulacijo: v njem se specificira in vpiše glavne podrobnosti prehoda v simulacijo:
    - Urnik simulacijske spremembe:
        - Začetek spremembe
        - Predviden konec spremembe – čas trajanja
    - *Assembly* verzija – iz konfiguracijskega sistema CMDB
    - Nosilci – izvajalci spremembe
    - Nadzorniki spremembe
    - Povezave na zahtevke za spremembo
    - Opis vpliva sprememb na poslovanje družbe
    - Natančno opredeljene predvidene prekinitve
    - Vzorci obveščanja strankam o (morebitnih) prekinitvah

  Potrditev tega dokumenta je potrebna s strani vodje oddelka tehnologije, ki je zahtevek podal in s strani vodje oddelka informacijske tehnologije, v katerem je sprememba implementirana.

Produkcijska sprememba

To je najpomembnejši sklop dokumentov pred izvedbo spremembe. Vsebuje torej glavne dokumente, na podlagi katerih se CAB odloči o potrditvi (ali zavrnitvi) produkcijske spremembe. ITIL predvideva različne tipe izdaj: izdaja sprememb (delta), polna izdaja, paketna izdaja in nujna izdaja. Če bi za naš primer naredili izdajo samo za nov WSDL, bi naredili izdajo sprememb. Ker pa je po navadi hkrati več sprememb znotraj enega sistema, delamo polno izdajo; še bolj pogosto pa paketno izdajo, saj poleg sistema, na katerem delam sam, vključujemo tudi na druge sisteme (programske produkte).

Za produkcijsko spremembo je seveda tudi potrebno odpreti hrošč, vsebuje pa naslednje dokumente:

- Plan implementacije in povrnitev v prvotno stanje: v ta dokument je razvijalec dolžan vpisati podrobnejši plan poteka spremembe v produkciji: uro začetka izvajanja spremembe, natančen popis aktivnosti implementacije produkcijske spremembe. Pomemben del dokumenta je tudi rubrika »Povrnitev v prvotno stanje«, kamor izvajalec vpiše postopek v primeru težav pri prehodu v produkcijo ali pa ob težavah/nepravilnem delovanju implementirane spremembe. Potrditev dokumenta je potrebna s strani drugega razvijalca in s strani nadrejenega.
- Zahtevek za produkcijo: v njem se specificira in vpiše glavne podrobnosti prehoda v simulacijo:
    - Urnik produkcijske spremembe:
        - Začetek spremembe
        - Predviden konec spremembe – čas trajanja
    - *Assembly* verzija – iz konfiguracijskega sistema CMDB
    - Nosilci – izvajalci spremembe
    - Nadzorniki spremembe
    - Povezave na zahtevke za spremembo
    - Natančno opredeljene predvidene prekinitve
    - Vzorci obveščanja strankam o (morebitnih) prekinitvah

  Potrditev tega dokumenta je potrebna s strani vodje oddelka tehnologije, ki je zahtevek podala in s strani vodje oddelka informacijske tehnologije, v katerem je sprememba implementirana.
- Post-produkcijski pregled: dokument, ki se izpolni naknadno, po opravljeni storitvi. Ima dve rubriki - pregled delovanja, pri kateri tehnični skrbnik po opravljeni spremembi preveri delovanje sistema in zabeleži ugotovitve in vsebinska potrditev implementacije, ki jo potrdi tehnolog – naročnik spremembe.

Tedenski CAB sestanki

Vsak teden je ob vnaprej določeni uri je na sporedu redni tedenski sestanek posvetovalnega tima (CAB), na katerem vodstvo združbe (direktorji sektorjev tehnologije in informacijske tehnologije), vodje oddelkov obeh prej omenjenih sektorjev ter posamezni zaposleni, odgovorni nosilci posameznih nalog oz. zahtevkov za spremembe, obravnavajo seznam zahtevkov za spremembo – produkcijske spremembe in jih potrjujejo. Seznam zahtevkov je avtomatsko pridobljen iz sistema Bugzilla.

Najprej odgovorni nosilec in/ali vodja oddelka odgovornega nosilca predstavi vsebino, ki jo zajema posamezna naloga oz. sprememba. Sledi predstavitev in analiza vpliva spremembe najprej v smislu vpliva znotraj sistema in na druge sisteme, potem pa še analiza vpliva na poslovanje združbe. Pomembno je pod nadzorom imeti predvsem prekinitve delovanja sistemov znotraj združbe, predvsem pa prekinitve za zunanje uporabnike storitev oz. stranke.

5.5.2.5 *Prehod v produkcijo*

Zadnja in najpomembnejša aktivnost tranzicije je seveda sam prehod v produkcijo. Ta se mora izvesti natančno tako, kot je opredeljeno v dokumentu Zahtevek za spremembo znotraj Produkcijske

spremembe. Kakršnokoli odstopanje ni zaželeno; če do njega vendarle pride, je to potrebno naknadno navesti v dokumentu Nadzor produkcije.

Po opravljeni tranziciji, navadno nekaj dni po uspešnem delovanju v produkcijskem okolju, se s strani naročnika vsebinske spremembe opravi vsebinski pregled produkcije in potrdi (ali zavrne) spremembo. Prav tako se s strani IT opravi tehnični pregled produkcije in potrdi (ali zavrne) spremembo. V primeru potrditev z obeh strani se izpolni dokument Pregled produkcije in zahtevek za spremembo se lahko formalno zapre.

5.5.3 Zaključek

Cilj podjetja, vpeljava standarda ISO/IEC 20000 zaradi zahteve ene od večjih strank je bila osnovana na implementaciji ITIL. S tem je podjetje pridobilo na dveh področjih: (deloma) je pridobilo certifikat ISO/IEC 20000, hkrati pa v svoje delovanje uvedlo dobro dokumentirane postopke iz ogrodja ITIL in s tem uvedlo dobre prakse, kakor je bilo prikazano tudi v fazi prenosa storitev (v produkcijo) in predvsem upravljanja sprememb. Upravljanje sprememb zahteva pripravo podrobne dokumentacije o vsem, kar je ključno pri prenosu novih izdaj programske in strojne ter druge opreme v obratovalno okolje. In to je tudi eden od bistvenih elementov ITIL – dobro dokumentirani in kontrolirani postopki.

6 Predstavitev standarda ISO/IEC 27001

Namen tega poglavja je podrobneje predstaviti standard ISO/IEC 27001, standard združbe ISO s področja informacijske varnosti. Standard ISO/IEC 27001, v nadaljevanju tudi krajše ISO/IEC 27001, je le eden od družine standardov ISO/IEC 27000, predstavljene v naslednjem razdelku.

6.1 Družina ISO/IEC 27000

ISO/IEC 27000 je serija standardov, ki ga je organizacija ISO rezervirala izključno za zadeve s področja informacijske varnosti, pri čemer je vsak od standardov napisan z določenim namenom in poudarkom. Izdani standardi iz družine 27000 so v pregled podani v tabeli 3.

Oznaka standarda	Opis
ISO/IEC 27000	Sistemi za upravljanje varovanja informacij (SUVI) (Pregled in slovar)
ISO/IEC 27001	Informacijska tehnologija - Tehnike varnosti - Sistemi za upravljanje informacijske varnosti varovanja informacij (SUVI) - Zahteve. ISO/IEC 27001:2005 se je skliceval na PDCA cikel; novejši ISO/IEC 27001:2013 se ne, vendar je bil posodobljen na druge načine, da odražajo spremembe v tehnologijah in v tem, kako združbe upravljajo z informacijami.
ISO/IEC 27002	Navodila in smernice za upravljanje informacijske varnosti
ISO/IEC 27003	Smernice za implementacijo sistema za upravljanje informacijske varnosti
ISO/IEC 27004	Upravljanje informacijske varnosti
ISO/IEC 27005	Upravljanje tveganj informacijske varnosti - merjenje
ISO/IEC 27006	Zahteve za organe, ki presojajo in certificirajo sisteme vodenja varovanja informacij
ISO/IEC 27007	Smernice za informacijsko varnost sistemov vodenja revidiranja (osredotočena na sistem upravljanja)
ISO/IEC TR 27008	Smernice za revizorje o nadzoru sistema upravljanja informacijske varnosti (osredotočen na informacije varnostnih kontrol)
ISO/IEC 27010	Upravljanje informacijske varnosti za medsektorske in med organizacijske komunikacije
ISO/IEC 27011	Smernice za upravljanje varnosti informacij za telekomunikacijske organizacije, ki temeljijo na standardih ISO/IEC 27002
ISO/IEC 27013	Smernice za celovito implementacijo ISO/IEC 27001 in ISO/IEC 20000-1
ISO/IEC 27014	Upravljanje informacijske varnosti
ISO/IEC TR 27015	Smernice za upravljanje varnosti informacij za finančne storitve
ISO/IEC 27018	Kodeks ravnanja za varstvo osebnih podatkov (PII) v javnih oblakih, ki deluje kot zavarovanja poklicne odgovornosti procesorjev
ISO/IEC 27031	Smernice za informacijske in komunikacijske tehnologije, pripravljene za neprekinjeno poslovanje
ISO/IEC 27032	Smernice za kibernetsko varnost
ISO/IEC 27033-1	Omrežna varnost - 1. del: Pregled in koncepti
ISO/IEC 27033-2	Omrežna varnost - 2. del: Smernice za načrtovanje in izvajanje varnosti omrežja
ISO/IEC 27033-3	Omrežna varnost - 3. del: scenariji Reference mreženje - Grožnje, tehnike oblikovanja in vprašanja nadzora
ISO/IEC 27033-5	Omrežna varnost - 5. del: Zagotavljanje komunikacije po omrežjih, ki uporabljajo navidezna zasebna omrežja (VPN)
ISO/IEC 27034-1	Varnost aplikacij - 1. del: Smernice za varnost aplikacij
ISO/IEC 27035	Upravljanje incidentov informacijske varnosti
ISO/IEC 27036-3	Informacijska varnost za dobavitelja razmerja - 3. del: Smernice za varnost informacijske in komunikacijske tehnologije dobavne verige

ISO/IEC 27037	Smernice za identifikacijo, zbiranje, pridobivanje in ohranjanje digitalnih dokazov
ISO 27799	Upravljanje informacijske varnosti v zdravstvu z uporabo standarda ISO/IEC 27002. Namen ISO 27799 je, da se zagotovi napotke zdravstvenim združbam in drugim nosilcem zdravstvenih osebnih podatkov o tem, kako zaščititi te podatke prek izvajanja ISO/IEC 27002

Tabela 3 - Pregled standardov družine ISO/IEC 27000
Vir: [58]

V nadaljevanju se v tem magistrskem delu osredotočam predvsem na ISO/IEC 27001, ki je specifikacija za sistem za upravljanje varovanja informacij in ISO/IEC 27002, ki je kodeks ravnanja, ki ponuja kontrole za reševanje varnostnih tveganj. Prav skupek obeh omenjenih in standarda ISO/IEC 27000 za področje informacijske varnosti opisujejo kot »skupen jezik za združbe po vsem svetu« [28].

6.2 Standard ISO/IEC 27001

Standard ISO/IEC 27001, trenutna verzija je iz leta 2013, je specifikacija za sistem za upravljanje varovanja informacij (SUVI). Cilj standarda je specificirati zahteve za vzpostavitev, implementacijo, delovanje, nadzor, pregled, vzdrževanje in izboljševanje sistema za upravljanje in varovanje informacij v združbi. Načrtovan je za zagotovitev izbora primernih varnostnih mehanizmov in kontrol za varovanje informacijskih sredstev. Prepoznan je kot mednarodno uveljavljena, strukturirana metodologija, namenjena upravljanju informacijske varnosti [20]. Standard vsebinsko zajema deset točk, ki opredeljujejo, kako naj bo SUVI načrtovan, udejanjen, nadzorovan, pregledan in izboljšan. Pregled teh točk je podan v tabeli 4.

Poglavje	Opis
1.	Kaj vse standard zajema
2.	Kako se dokument sklicuje
3.	Ponovna uporaba izrazov in definicij v ISO/IEC 27000
4.	Organizacijski kontekst in interesne skupine (deležniki)
5.	Vodenje informacijske varnosti in podpora politiki na višjih nivojih
6.	Načrtovanje sistema upravljanja varovanja informacij, ocena tveganja, soočanje s tveganjem
7.	Podpora sistemu upravljanja varovanja informacij
8.	Izvedba operativnega sistema za upravljanje varovanja informacij
9.	Pregled uspešnosti sistema
10.	Izboljševalni ukrepi

Tabela 4 - Poglavja standarda ISO/IEC 27001
Vir: [59]

V nadaljevanju je v tabeli 5 navedenih 14 skupin, ki združujejo 114 kontrol, ki se uporabljajo kot orodje za upravljanje varnosti, kot nekakšen seznam kontrol, s katerimi izboljšujemo informacijsko varnost. Standard ISO/IEC 27001 te kontrole vodi kot dodatek A. Pri vsaki oznaki skupine je podan kratek (vsebinski) opis skupine.

Oznaka	Vsebina
A.5	Politike informacijske varnosti - kako naj bodo politike spisane in pregledane
A.6	Organizacija informacijske varnosti - kako naj bodo porazdeljene odgovornosti
A.7	Varnost človeških virov - kontrole, ki se uporabljajo pred, med ali po zaposlitvi

A.8	Upravljanje premoženja - kontrole, povezane z inventarjem sredstev in njihovo primerno uporabo; uporabljajo se tudi za klasifikacijo informacij in delo z mediji oziroma nosilci informacij
A.9	Nadzor dostopa in upravljanje dostopa uporabnikov - uporablja se za politiko pristopnih pravic, upravljanje uporabniških pravic, dostope do sistemov in aplikacij in uporabniške odgovornosti
A.10	Kriptografske tehnologije - kontrole za kriptografijo in delo s ključi
A.11	Fizično varovanje prostorov in opreme združbe - kontrole, ki opredeljujejo varnostna področja, kontrole dostopov, varovanje pred grožnjami, varnost opreme, uničevanje dokumentacije, politiko čiste mize in čistega ekrana itn.
A.12	Operativna varnost - kontrole za upravljanje produkcije: upravljanje sprememb, upravljanje kapacitet, neželena programska oprema, varnostne kopije, logiranje, nadzor, namestitve, ranljivosti itn.
A.13	Varne komunikacije in prenos podatkov - mrežna varnost, segregacija, mrežne storitve, prenos informacij, sporočila itn.
A.14	Varno pridobivanje, razvoj in podpora informacijskih sistemov - kontrole, ki opredeljujejo varnostne zahteve in varnost v razvojnih procesih, pa tudi procesih podpore, npr. uporabnikom
A.15	Varnost za dobavitelje in tretje osebe - kontrole, ki opredeljujejo kaj naj vsebujejo pogodbe in kako nadzorovati dobavo
A.16	Obvladovanje incidentov - kontrole za poročanje o dogodkih in incidentih, opredelitev odgovornosti, procedure za odgovore in zbirko dokazov
A.17	Vidiki informacijske varnosti upravljanja neprekinjenega poslovanja - kontrole, ki zahtevajo načrtovanje kontinuitete, procedur, verifikacije in pregled poslovanja
A.18	Skladnost z notranjimi zahtevami, kot so politike, in z zunanjimi zahtevami, kot so zakoni - za identifikacijo veljavnih zakonov in predpisov, zaščito intelektualne lastnine, varovanje osebnih podatkov in pregled informacijske varnosti

Tabela 5 - Pregled seznama kontrol iz dodatka standarda ISO/IEC 27001:2013
Vir: [59]

Obravnavani standard lahko štejemo za tehnološki izdelek, ki pokriva sistematično znanje z navodili, kako naj bo v združbah upravljana varnost [60].

ISO/IEC 27001 se je razvijal potopoma na podlagi prejšnjih enako pomenskih standardov. Začetki segajo v leto 1993, ko je organizacija *British professional association, the National Computing Centre (NCC)* izdala dokument z naslovom *PD 0003 Kodeks ravnanja pri upravljanju informacijske varnosti*. Kmalu zatem ga je organizacija *The British Standards Institute (BSI)* prevzela in v letu 1995 izdala standard *BS 7799-1 IT – Varnostne tehnike – Kodeks ravnanja za upravljanje informacijske varnosti*. Drugi del, *BS 7799-2 sistemi za upravljanje informacijske varnosti – Specifikacija in navodila za uporabo* je omogočila združbam certificirati njihove procese. Organizacija ISO je uskladila te standarde še z ISO 9001 in razvila ISO/IEC 27001, ki ga je izdala oktobra leta 2005 [28].

Leta 2013 si je 22293 združb iz 105 držav lastilo certifikat ISO/IEC 27001 [60]. Standard uporabljajo mala, srednja in velika podjetja in je tako fleksibilen, da se ga lahko prilagodi vsakemu tipu podjetja [61]. Glede na področja, v katerem delujejo podjetja s pridobljenim certifikatom ISO/IEC 27001, podatki iz leta 2013 kažejo, da jih je pričakovano največ s področja informacijske tehnologije, sledi skupina več področij z manjšo zastopanostjo, potem pa sledi področje gradnje, transporta in telekomunikacij, električne in optične opreme itd. [60], če naštejem le prvih pet področij.

Združbe imajo za uporabo standarda na voljo dve možnosti. Lahko sprejmejo standard brez opravljanja certifikacije, kar bi pomenilo, da le koristijo najboljše prakse, ki jih standard ponuja, ali pa se odločijo za to, da jih pregleda akreditirani revizor in pridobi certifikat. Revizija ponovi obisk certificirane združbe glede na načrtovan interval pri čemer se preveri, ali sistem za upravljanje

varovanja informacij deluje in ali se podjetje glede tega nenehno izboljšuje. V primeru večjih anomalij se certifikat lahko ukine oziroma začasno vzame [60].

6.3 Standard ISO/IEC 27002

Standard ISO/IEC 27002 podaja smernice oziroma navodila implementacije za nanizanih 14 zahtev, na kratko predstavljenih v prejšnjem razdelku, od A.5. do A.18 iz standarda ISO/IEC 27001. Za razliko od slednjega, za standard ISO/IEC 27002 ne moremo pridobiti certifikata, saj ne gre za standard upravljanja, ki opredeli, kako naj določen sistem deluje in pri ISO/IEC 27001 gre ravno za to: opredeljuje, kakšen naj bo sistem za upravljanje varovanja informacij (SUVI) [28]. Kontrole, ki so zajete v ISO/IEC 27002, so poimenovane enako kakor skupine kontrol v dodatku oziroma prilogi ISO/IEC 27001, razlika je v nivoju podrobnosti. Vsaka od kontrolnih točk je v ISO/IEC 27001 napisana v stavku ali dveh, v ISO/IEC 27002 pa razložena veliko podrobneje z navodili za njihovo implementacijo.

7 Podrobna primerjava in integracija ter implementacija v praksi

7.1 Zakaj integracija

Pri svoji raziskavi sem sledil hipotetičnemu vprašanju, kako lahko podjetje zmanjša stroške in časovni okvir pri morebitni potrebi po certifikatu standarda ISO/IEC 27001 in pri tem uporabi čim več že implementiranega iz zahtev PCI DSS in ogrodja ITIL. Pred leti je bila združba že na tem, da udejanji standard ISO/IEC 27001, pa se je zaradi različnih razlogov ta aktivnost prekinila. Trenutno standarda ISO/IEC 27001 podjetje (še) ne potrebuje, se pa je v preteklosti že nakazalo, da ga bo morda potrebovalo, kot je npr. zahteva stranke ipd.

Čeprav so varnostna ogrodja in standardi v skladu z zamislimi tistih, ki so jih razvili, gre vseeno za splošna ogrodja, primerna tudi za združbe drugačnih tipov. Zaradi tega je enaka osnovna načela ogrodja možno uporabiti na več standardov, regulativ ali zahtev industrije [30]. To torej pomeni, da imajo ogrodja in standardi (lahko) veliko skupnih enako pomenskih načel.

Poleg omenjene možnosti, da je od podjetja s strani kakšne zunanje entitete zahtevano, da pridobi certifikat ISO/IEC 27001, se podjetje lahko za to odloči samoiniciativno. Integracija različnih standardov in ogrodij namreč lahko prinese veliko prednosti:

- Zmanjšanje stroškov za dosego določenega standarda, ki ga združba potrebuje ali se od nje zahteva [62]. Zmanjšanje stroškov virov tako strojne kot programske opreme.
- Določen del standarda, za katerega je podjetje že certificirano, se pokriva z vsebino standarda, ki ga združba želi osvojiti. Tako se torej nek proces, ki je že v veljavi, le ponovno uporabi ali deloma modificira.
- Vključitev bistvenih elementov enega standarda v elemente drugega standarda lahko le-te vsebinsko dopolnjuje in izboljšuje ter povečuje njihovo učinkovitost in uspešnost.
- Vsak od standardov ima lahko določene pomanjkljivosti, ki se jih lahko dopolni z zahtevami in vsebino drugega standarda; tako imajo lahko združbe najboljše iz dveh (ali več) standardov [8].
- Zmanjšanje nivoja tveganj [62] (za neuspeh pri implementaciji ipd).

Vsekakor pa se lahko najdejo tudi slabosti uporabe več kot enega (sorodnega) standarda hkrati. Lahko se zgodi, da pride do nerešljivo podvojenih zahtev, pa tudi do medsebojnih konfliktov med zahtevami. Takšno stanje lahko v združbi privede do slabo izkoriščenih virov za udejanjenje zahtev kot tudi do redundančnih uporabljenih komponent SUVI, zato je smiselno do potankosti razumeti razmerja med zahtevami iz različnih standardov [63].

V naslednjih nekaj odstavkih so podani nekateri razlogi in utemeljitve, zakaj narediti integracijo več različnih SUVI standardov.

Ataya [40] tako podaja primer ponovne uporabe oziroma integracije PCI DSS standarda v že obstoječ SUVI. Zrele združbe, kot pravi, ki udejanjijo SUVI, kot so ISO/IEC 27001, COBIT ali ITIL, lahko pristopijo k PCI DSS z načinom z vrha navzdol (*top-down approach*). Pri tem načinu gre za dodajanje zahtev PCI DSS na drug, že obstoječ cilj informacijske varnosti in skladnosti. Aktivnosti zaščite in skladnosti se nato vgradijo v obstoječe okolje in tako imamo korist varovanja in praks, ki so že v veljavi. Elementi, ki tvorijo omenjeno okolje, so ljudje, tehnologija, procesi in združba. Soodvisnosti med temi elementi so del sistematskega pristopa, ki jih morajo razumeti tisti, ki so odgovorni za aktivnosti skladnosti [BMIS 2010 v [40]].

Nicho [41] trdi, da PCI DSS, ki se v glavnem osredotoča le na podatke o imetnikih plačilnih kartic, ni dovolj za popolno zaščito sredstev informacijskih sistemov, saj je za le-to potreben celovit pregled nad informacijsko varnostjo na podlagi varnostne in revizijske skladnosti, ki jih ponujajo določena ogrodja, ki nudijo celovit varnostni sloj nad podatki združbe. Zato predlaga, da združbe skupaj s PCI

DSS integrirajo še kakšnega od varnostnih ogrodij in s tem naredijo celovito zaščito podatkov o imetnikih plačilnih kartic.

7.2 Medsebojna primerjava obravnavanih standardov

V uvodu tega razdelka je najprej smiselna opomba. Kot sem že pisal v poglavju 5, je ITIL ogrodje, s katerim lahko dosežemo (tudi) implementacijo standarda ISO/IEC 20000, v razdelku 6.3 pa, da je ISO/IEC 27001 standard, za katerega podjetje lahko pridobi certifikat in ISO/IEC 27002 zbirka navodil in praks, kako zahteve iz ISO/IEC 27001 udejanjiti. V literaturi se torej zaradi sorodnosti teh standardov/ogrodij pojavljajo primerjave različnih kombinacij med ISO/IEC 27001/27002 in ITIL/ISO/IEC 20000, pa tudi teh dveh z ostalimi standardi. Prav zato bom tudi sam primerjal in povzemal primerjave na tak način.

Vsak od obravnavanih in primerjanih standardov v tem magistrskem delu ima svoj obseg in glavni cilj. Tako PCI DSS obsega varovanje predvsem informacij in podatkov o debetnih, kreditnih, predplačniških, spletnih ATM in POS transakcijah, ITIL pa je namenjen obvladovanju storitev podjetja, ISO/IEC 27001 pa varovanju informacij in gradnji sistema za upravljanje varovanja informacij.

Naslednja preglednica (Tabela 6) poleg v prejšnjem odstavku omenjenega obsega prikazuje nekaj najbolj osnovnih izbranih lastnosti obravnavanih standardov. Te lastnosti so kategorija, poudarek (glavna usmeritev in namen standarda), paradigma, na kateri standard sloni, obseg standarda, ki ga pokriva, struktura, organizacijski model ter informacija o tem, ali je za standard/ogrodje možno pridobiti certifikat. Precej od naštetega je bilo sicer omenjeno že v prejšnjih treh poglavjih, ker pa gre v tem poglavju za podrobnejšo primerjavo standardov, je koristno te lastnosti prikazati še v razpredelnici.

	PCI DSS	ITIL	ISO/IEC 27001
Kategorija	Standard	Ogrodje	Standard
Poudarek	Udejanjanje varnostnih kontrol s poudarkom na varovanju podatkov o imetnikih plačilnih kartic	Zagotavljanje IT storitev združbe	Udejanjenje varnostnih kontrol s poudarkom na upravljanju tveganj
Paradigma	Varnost podatkov o imetnikih plačilih kartic (CDS)	ITSM	SUVI (ISMS)
Obseg	Varnost podatkov in informacij transakcij kartičnega poslovanja	Upravljanje storitev združbe	Smernice za IT varnost v združbi
Struktura	12 krovnih poglavij, razdeljenih v zahteve	5 krovnih področij, razdeljenih na procese	10 poglavij, 14 zahtev iz dodatka A
Organizacijski model	Vodstvo, oddelek za informacijsko varnost	Vsi deležniki	Vodstvo, oddelek za informacijsko varnost
Certifikacija	DA	NE	DA

Tabela 6 - Pregled lastnosti obravnavanih ogrodij

Kot je vidno iz preglednice, imata več enakih oziroma sorodnih lastnosti (pričakovano) PCI DSS in ISO/IEC 27001. Tako imata skupno kategorijo (standard) in organizacijski model (pripada predvsem vodstvu in oddelku za informacijsko varnost). Poudarek je soroden PCI DSS in ISO/IEC 27001, to je implementacija varnostnih kontrol, vendarle gre pri PCI DSS za varovanje podatkov o imetnikih plačilnih kartic, pri ISO/IEC 27001 pa je poudarek na upravljanju tveganj. Poudarek ITIL je na zagotavljanju upravljanja storitev združbe. Naj omenim še paradigmo: paradigma PCI DSS je varnost

podatkov o imetnikih plačilnih kartic (*CDS – Card Data Security*), paradigma ITIL je upravljanje IT storitev (*ITSM – IT Service Management*), paradigma ISO 27001 pa sistem za upravljanje varnosti informacij (*ISMS – Information Security Management System*).

V poglavju 3.2.2 so bili omenjeni in opisani kriteriji 11EC, ki naj bi v popolnosti pokrili vse vidike sistema za upravljanje varovanja informacij (SUVI). V tabeli 7 je razvidna vključenost posameznega merila v posameznem standardu. Tako PCI DSS in ISO/IEC 27001 celoti izpolnjujeta vseh 11 meril, med tem ko jih ITIL pokrije le deloma. Tako ostajajo neizpolnjeni upravljanje komunikacij in delovanja, pridobitev, razvoj in vzdrževanje informacijskega sistema, varnost človeških virov ter fizična varnost in varnost okolja, v katerem združba deluje.

Merilo 11EC	PCI DSS	ITIL - ISO/IEC 20000	ISO/IEC 27001
1. Politika informacijske varnosti	DA	DA	DA
2. Upravljanje komunikacij in delovanja	DA	NE	DA
3. Nadzor dostopa	DA	DA	DA
4. Pridobitev, razvoj in vzdrževanje informacijskega sistema	DA	NE	DA
5. Organizacija informacijske varnosti	DA	DA	DA
6. Upravljanje sredstev	DA	DA	DA
7. Upravljanje incidentov s področja informacijske varnosti	DA	DA	DA
8. Upravljanje neprekinjenega poslovanja	DA	DA	DA
9. Varnost človeških virov	DA	NE	DA
10. Fizična varnost in varnost okolja	DA	NE	DA
11. Skladnost	DA	DA	DA

Tabela 7 - Pregled podprtosti standardov po merilih 11EC
Vir: [5]

V nadaljevanju v vsakem razdelku posebej primerjam vsakega od treh obravnavanih standardov v parih in sicer glede vsebino na visokonivojskih zahtevah, pri čemer so lahko podrobneje omenjene in opisane lastnosti iz prejšnjih poglavij in tega razdelka.

7.2.1 Primerjava ITIL (ISO/IEC 20000) in ISO/IEC 27001

ISO/IEC 27001/ISO/IEC 27002 in ITIL sta komplementarna oziroma se dopolnjujeta in pri obeh naj bi bil poglavitni namen iskanje najboljših praks, pri čemer je ITIL namenjen najboljšim praksam upravljanja storitev, ISO/IEC 27001/ISO/IEC 27002 pa najboljšim praksam informacijske varnosti. Oba slonita na PDCA modelu [62].

Gehrmann [64] v svojem članku primerja ISO/IEC 27002 in ITIL, kjer zapiše, da je le-ta namenjen problemom in izzivom informacijske varnosti in ne le upravljanju IT ter da s temi splošnimi cilji tako kot ekvivalent ne odgovarja ITIL metodologiji, kot se to lahko primerja ITIL in COBIT. Saint-Germany (2005) v [64] ugotavlja, da udejanjenje varnosti in kontrol iz ISO/IEC 27002 v kombinaciji z ITIL zmanjšujejo kritične nevarnosti, ki lahko vplivajo na rezultate določenega projekta.

Struktura ISO/IEC 27002 naj bi bila aplicirana glede na združbo in tako zagotavljala vsesplošno varnost na vseh nivojih informacijske varnosti v podjetju. Problematika administracije in upravljanja, ki jo obravnava ITIL, nima ekvivalentne strukture v ISO/IEC 27002. ISO/IEC 27002 ima funkcije za ohranitev zaupnosti, celovitosti in razpoložljivosti informacij v organizacijah. Ta razpoložljivosti informacij se z vidika kakovosti, zanesljivosti in vzdrževanja IT obdeluje v ITIL. Simonsson in Johnson (2008) v [64] poudarjata, da lahko ISO/IEC 27002 skupaj z ITIL pomaga oblikovati procese,

povezane z dostavo in podporo IT. V nadaljevanju Gehrmann znotraj teh dveh standardov primerja še finančni vidik, ki ga ISO/IEC 27002 ne obravnava celostno. Ukvarja se samo z obvladovanjem tveganj, pri čemer odločitve o tveganjih sprejema ravnatelj. Ta pristop, ki se v ITIL obravnava drugače, ponuja učinkovito upravljanje tveganj in s financami povezanih stroškov. Gehrmann predlaga, da bi bila ITIL metodologija uporabljena za opredelitev strategij, konceptov in procesov, ki so povezani z upravljanjem informacijske tehnologije. ISO/IEC 27002 pa naj se v združbah v povezavi z upravljanjem informacijske tehnologije uporabi za vprašanja IT varnosti.

Ravno za slednje, torej za vprašanja informacijske varnosti, sem po pregledu procesov, ki jih nudi ITIL za obvladovanje storitev in predlogih iz članka *Integration of Information Security Essential Controls into Information Technology Infrastructure Library – A Proposed Framework* [65], pripravil seznam procesov iz različnih področij, ki so tesno povezani z informacijsko varnostjo. Ti procesi so v pregled na voljo v tabeli 8 spodaj.

Procesi ITIL	Področje ITIL, kateremu pripada proces
Upravljanje na nivoju storitev	Načrtovanje storitev
Upravljanje zmožnosti in kapacitet	Načrtovanje storitev
Upravljanje razpoložljivosti	Načrtovanje storitev
Upravljanje kontinuitete IT storitev	Načrtovanje storitev
Upravljalni sistem za informacijsko varnost	Načrtovanje storitev
Upravljanje dobaviteljev	Načrtovanje storitev
Upravljanje sprememb	Prenos storitev
Upravljanje incidentov	Delovanje storitev
Izpolnjevanje zahtev	Delovanje storitev
Upravljanje problemov	Delovanje storitev

Tabela 8 - Identificirani procesi ITIL z vsebino varovanja informacij s pripadajočimi področji

Mubashir Ali in ostali [9] ugotavljajo, da lahko ISO/IEC 27002 pomaga ITIL pri funkcijah pomoči uporabnikom (*help desk*), konfiguraciji incidentov in problemov, upravljanju sprememb in izdaj, pa tudi upravljanju financ ter SLA in upravljanju neprekinjenega poslovanja. Ugotavljajo še, da ISO/IEC 27002 lahko v ITIL, ki kot tak izboljša IT procese in kontrole, dodamo varnost v te procese in kontrole.

Zrel sistem za upravljanje storitev, ki ga nudi ITIL, lahko koristno pomaga pri doseganju kontrol, ki podpirajo sistem za upravljanje varnosti informacij (SUVI) [66].

7.2.2 Primerjava ISO/IEC 27001 in PCI DSS

Standarda s področja informacijske varnosti PCI DSS in ISO/IEC 27001 in njuna primerjava med zahtevami in vsebino je verjetno (izmed obravnavanih standardov/ogrodij) najbolj preprosta. Lahko bi rekli, da gre pri ISO/IEC 27001 za skladnost procesov, pri PCI DSS pa za skladnost pretoka podatkov. Wright [67] pravi, da PCI DSS in ISO/IEC 27001 nista bistveno različna v zahtevah glede varnosti podatkov.

V nadaljevanju podajam podobnosti in razlike med standardoma [8]. Podobnost med njima je v tem, da oba standarda, da združba ostane skladna in certificirana, zahtevata (vsakoletne) ponovitve revizij. Več je razlik. PCI DSS je večinoma priznan v Severni Ameriki in Evropi, skladnost po standardu je obvezna, pri čemer morajo biti izpolnjene vse zahteve. Ločitev sistemov je visoka, fleksibilnost je nizka, ni omembe kakršnih koli predhodnih zahtev za upravljanje ogrodja in PCI DSS velja predvsem za podatke o imetnikih plačilnih kartic. Na drugi strani je ISO/IEC 27001 mednarodno priznan ter prostovoljen pri odločitvi za združbe, ali opravljajo skladnost z njim. Ločitev sistemov je nizka, stopnja prilagodljivosti je visoka in je le malo podrobnosti o tem, kako se nadzor dejansko izvaja.

Tudi obseg standardov je različen. Medtem, ko je obseg ISO/IEC fleksibilen in si ga podjetje izbere samo, gre pri PCI DSS s tega vidika za bolj tog standard, obseg so preprosto podatki o imetnikih plačilnih kartic [6]. S tega vidika je, zaradi fleksibilnosti, za združbe lažje ustrezati ISO/IEC 27001 kot PCI DSS.

PCI DSS 3.1	ISO/IEC 27001:2013
Requirement 1: Install and Maintain a Firewall Configuration to Protect Cardholder Data	A.12 Operations security A.13 Communications security
	A.9.1.2 Access to networks and network services
Requirement 2: Do Not Use Vendor-Supplied Defaults for System Passwords and Other Security Parameters	A.12 Operations security A.13 Communications security
	A.9 Access control
	A.14 System acquisition, development and maintenance
Requirement 3: Protect Stored Cardholder Data	A.12 Operations security A.13 Communications security
	A.14 System acquisition, development and maintenance
	A.18 Compliance
Requirement 4: Encrypt Transmission of Cardholder Data Across Open, Public Networks	A.12 Operations security A.13 Communications security
	A.9 Access control
Requirement 5: Use and Regularly Update Anti-Virus Software	A.12.2 Protection from malware
Requirement 6: Develop and Maintain Secure Systems and Applications	A.12 Operations security A.13 Communications security
	A.9 Access control
	A.14 System acquisition, development and maintenance
Requirement 7: Restrict Access to Cardholder Data by Business Need-to-Know	A.6.1.1 Information security roles and responsibilities
	A.9.2.6 Removal or adjustment of access rights
	A.9 Access control
Requirement 8: Assign a Unique ID to Each Person with Computer Access	A.7 Human resource security
	A.12 Operations security A.13 Communications security
	A.9 Access control
Requirement 9: Restrict Physical Access to Cardholder Data	A.7 Human resource security
	A.11 Physical and environmental security
	A.12 Operations security A.13 Communications security
Requirement 10: Track and Monitor All Access to Network Resources and Cardholder	A.12 Operations security A.13 Communications security
	A.9 Access control

Requirement 11: Regularly test security systems and processes	A.12 Operations security A.13 Communications security
	A.14 System acquisition, development and maintenance
Requirement 12: Maintain a Policy that Addresses Information Security	All [15]

Tabela 9 - Visokonivojska preslikava zahtev vsebinsko enakega pomena med standardoma PCI DSS in ISO/IEC 27001:2013
Vir: [7]

PCI DSS 3.1 – ISO/IEC 27001	A .5	A .6	A .7	A .8	A .9	A. 10	A. 11	A. 12	A. 13	A. 14	A. 15	A. 16	A. 17	A. 18
Requirement 1: Install and Maintain a Firewall Configuration to Protect Cardholder Data					X				X					
Requirement 2: Do Not Use Vendor-Supplied Defaults for System Passwords and Other Security Parameters					X			X	X	X				
Requirement 3: Protect Stored Cardholder Data								X	X	X				X
Requirement 4: Encrypt Transmission of Cardholder Data Across Open, Public Networks					X			X	X					
Requirement 5: Use and Regularly Update Anti-Virus Software								X						
Requirement 6: Develop and Maintain Secure Systems and Applications								X	X					
Requirement 7: Restrict Access to Cardholder Data by Business Need-to-Know		X			X									
Requirement 8: Assign a Unique ID to Each Person with Computer Access			X		X			X	X					
Requirement 9: Restrict Physical Access to Cardholder Data			X				X	X	X					
Requirement 10: Track and Monitor All Access to Network Resources and Cardholder					X			X	X					
Requirement 11: Regularly test security systems and processes								X	X	X				
Requirement 12: Maintain a Policy that Addresses Information Security	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Tabela 10 - Preslikava (visokonivojskih) zahtev enakega pomena med standardoma PCI DSS in ISO/IEC 27001:2013, drugi pogled
Vir: [7]

V tabeli 10 je prikazana preslikava zahtev med standardoma PCI DSS in ISO/IEC 27001, povzetem po [7] in posodobljenem na najnovejšo verzijo obeh standardov (PCI DSS verzija 3.1 iz aprila 2015 in ISO/IEC 27001:2013, izdan 25. 9. 2013). Za posodobitev zahtev ISO/IEC 27001 sem uporabil na spletu prosto dostopen dokument, ki za vsakega od razdelkov iz verzije 2005, pri katerem je prišlo do spremembe, preslika v razdelek iz aktualne verzije 2013 [68]. Naj poudarim, da je v tabeli 9 preslikava manj natančna, gre predvsem za površen prikaz vsebinsko deloma ali v celoti vsebinskih enako pomenskih zahtev. Bolj natančna preslikava bo prikazana v nadaljevanju. Še en pogled na preslikavo zahtev med standardoma je podan v tabeli 10.

Naj za konec tega razdelka podam trditev [67], da je PCI DSS odličen tehnični standard, ki pa vendarle potrebuje SUVI ravno zaradi tega, da ga lahko upravlja.

7.2.3 Primerjava PCI DSS in ITIL

Security Standards Council, avtorji standarda PCI DSS, sami navajajo [69], da lahko z uporabo ostalih standardov in ogrodij ter dobrih praks v kombinaciji s PCI DSS pomembno dopolnimo in izboljšamo učinkovitost varovanja podatkov o imetnikih plačilnih kartic. To trditev bi lahko razširili tudi na ostala področja varnosti. Navajajo še [69], da ITIL lahko podpira aktivnosti za PCI skladnost, ki so v teku. ITIL poudarja tudi stalno spremljanje ključnih poslovnih procesov in formalnega upravljanja sprememb s čimer minimizira prekinitve v poslovanju oziroma incidente ter naredi varnostne ocene kar del vsakdanjika.

7.3 Model poenostavljene implementacije ISO/IEC 27001 z uporabo PCI DSS in ITIL

Namen tega razdelka je pregledati in poiskati zahteve standarda ISO/IEC 27001, ki bi jih podjetje lahko pokrilo s tem, kar trenutno že ima na voljo iz standarda PCI DSS in ogrodja ITIL. Pri svojem delu sem izhajal iz tega, da bi podjetje pokrilo ISO/IEC 27001 standard v celoti.

7.3.1 Preslikava zahtev ISO/IEC 27001:2013 v PCI DSS 3.1

V prejšnjem poglavju je bilo ugotovljeno, da se standarda v svojih zahtevah precej prekrivata in v tem razdelku podajam čisto konkreten pregled vsake od zahtev, izdelek tega pregleda pa je preglednica v poglavju 10.2 v dodatku, v kateri je narejena preslikava zahtev trenutno najnovejših standardov ISO/IEC 27001 (izdaja 2013) iz dodatka A z zahtevami, ki jih narekuje PCI DSS (verzija 3.1). Potrebno je bilo do potankosti razumeti določena zahteve tako enega kot drugega standarda, s čimer je bilo potem lažje najti ekvivalentne zahteve med enim in drugim standardom. Preslikava je bila narejena na podlagi preslikave ISO/IEC 27001:2005 in PCI DSS 2.0 [70], ki jo je naredil Matthias Hofherr. Dokument je prosto dostopen na svetovnem spletu in do zahteve natančno podaja pregled enakovrednih oziroma enako pomenskih zahtev iz obeh standardov. Preden sem aktualiziral preslikavo na najnovejše verzije obeh standardov, sem pregledal obstoječe in ni bilo zaznati potrebe po kakšni večji spremembi; preslikava je dobro narejena. Aktualizacija je torej potekala tako, da sem najprej pripravil seznam zahtev ISO/IEC 27001 iz leta 2013, nato pa zanje poiskal zahteve, ki so vsebinsko enake za ISO/IEC 27001 iz leta 2005. Za ta namen sem poiskal dokument, ki je prosto dostopen na spletu [68]. Nekaj zahtev je ostalo enakih, večina zahtev je vsebinsko ostala enaka, spremenilo se je samo številčenje, nekaj zahtev je odstranjenih, nekaj pa je novih. Na podlagi novo pridobljenega seznama sem uporabil omenjeno preslikavo [70] in tako v preglednici za vsako ISO/IEC 27001:2013 zahtevo pripisal zahtevo PCI DSS 2.0. Za končen pregled je bilo potrebno preslikati še zahteve PCI DSS 2.0 v najnovejšo verzijo 3.1, ki sem jo naredil preko vmesnega standarda 3.0. Med standardoma 2.0 in 3.0 je nekaj razlik, verziji 3.0 in 3.1 pa sta skoraj enaki. Dokumentacija s spremembami PCI DSS je na voljo na uradni spletni strani standarda PCI DSS [71], [72].

V nadaljevanju v tabeli 11 podajam seznam zahtev ISO/IEC 27001:2013, ki so nove in za katere je bilo potrebno najti ustrezne zahteve iz PCI DSS 3.1, ki so podane na desni strani tabele.

ISO/IEC 27001:2013	PCI DSS 3.1
A.6.1.5 Information security in project management	[Many]
A.12.6.2 Restrictions on software installation	12.3.3 A list of all such devices and personnel with access 12.3.7 List of company-approved products
A.14.2.1 Secure development policy	6.7 Ensure that security policies and operational procedures for developing and maintaining secure systems and applications are documented, in use, and known to all affected parties.
A.14.2.5 Secure system engineering principles	/

A.14.2.6 Secure development environment	6.4.2 Separation of duties between development/test and production environments 6.4.3 Production data (live PANs) are not used for testing or development 6.4.4 Removal of test data and accounts before production systems become active
A.14.2.8 System security testing	11.3 Implement a methodology for penetration testing that includes the following: 11.3.1 Perform *external* penetration testing at least annually and after any significant infrastructure or application upgrade or modification 11.3.2 Perform *internal* penetration testing at least annually and after any significant infrastructure or application upgrade 11.3.3 Exploitable vulnerabilities found during penetration testing are corrected and testing is repeated to verify the corrections. 11.3.4 If segmentation is used to isolate the CDE from other networks, perform penetration tests
A.15.1.1 Information security policy for supplier relationships	9.9.3 Provide training for personnel to be aware of attempted tampering or replacement of devices. Training should include the following:
A.15.1.3 Information and communication technology supply chain	9.9.3 Provide training for personnel to be aware of attempted tampering or replacement of devices. Training should include the following:
A.16.1.4 Assessment of and decision on information security events	10.3 Record at least the following audit trail entries for all system components for each event: 10.6.1 Review the following at least daily: 10.6.2 Review logs of all other system components periodically based on the organization's policies and risk management strategy, as determined by the organization's annual risk assessment. 10.6.3 Follow up exceptions and anomalies identified during the review process. 12.10 Implement an incident response plan. Be prepared to respond immediately to a system breach. 12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:
A.16.1.5 Response to information security incidents	12.10 Implement an incident response plan. Be prepared to respond immediately to a system breach. 12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:

A.17.2.1 Availability of information processing facilities	9.5 Physically secure all media. 9.5.1 Store media backups in a secure location, preferably an off-site facility, such as an alternate or backup site, or a commercial storage facility. Review the location's security at least annually. 12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:

Tabela 11 - Novo prepoznane zahteve standarda ISO/IEC 27001:2013 in ustrezne zahteve iz PCI DSS 3.1

Posebna zahteva iz seznama iz tabele 11 je *A.14.2.5 Secure system engineering principles*, ki zapoveduje, da naj bodo uvedena, dokumentirana, vzdrževana in uporabljana načela za inženiring varnih sistemov za implementacijo kakršnega koli informacijskega sistema. Ker je zahteva podana precej ohlapno, lahko najdemo enako pomenske zahteve iz PCI DSS pri skoraj vseh poglavjih.

Celotna tabela, ki jo lahko kot pomoč uporabimo pri iskanju enako pomenskih zahtev med standardoma (v nadaljevanju ju bom za predlog modela implementacije standarda ISO 27001 in identificirane primere tveganj uporabil tudi sam), je v celoti na voljo v dodatku 10.2 (Preslikava zahtev med ISO/IEC 27001 Annex A in PCI DSS 3.1).

7.3.2 Preslikava zahtev ISO/IEC 27001 v ITIL

Primerjavo v zahtevah med ISO/IEC 27001 in ITIL povzemam po [73], kjer je sicer predstavljena primerjava med ISO/IEC 27001 in ISO/IEC 20000, sam pa sem zahteve ISO/IEC 20000 poiskal v ITIL. Kot je videti v tabeli 12, se najde dosti sorodnosti med ITIL in ISO/IEC 27001, obstajajo namreč varnostne kontrole dodatka A standarda ISO/IEC 27001, ki se lahko upravljajo kot procesi v ITIL.

ISO/IEC 27001	ITIL
A.12.1.2 Change management	Change management (Service transition)
A.12.1.3 Capacity management	Capacity management (Service design)
A.15 Supplier relationships	Supplier management (Service design)
	Service level management (Service design)
A.16 Information security incident management	Incident management (Service operation)
	Request fulfillment (Service operation)
	Problem management (Service operation)
A.17 Information security aspects of business continuity management	Service continuity management (Service design)
	Availability management (Service design)

Tabela 12 - Visokonivojska preslikava zahtev/procesov med standardom ISO/IEC 2701 in ITIL
Vir: [73]

Na splošno velja, da ITIL (oziroma ISO/IEC 20000) vsebinsko pokriva več kot ISO/IEC 27001, slednji pokriva le področje varnosti. Če se osredotočimo na posamezne zahteve, je tako npr. poglavju A.12.1.2 Change management - upravljanje sprememb iz ISO/IEC 27001 ekvivalent proces Upravljanje sprememb iz področja Prenosa storitev v ITIL, pri čemer slednje pokriva več vsebine. Zahteva A.12.1.3 Capacity management – upravljanje kapacitet je soroden procesu upravljanja sprememb iz področja Načrtovanja storitev v ITIL: tudi tu slednji pokriva več vsebine. Še zadnji primer, ki ga omenjam, je poglavje A.16 Information security incident management iz standarda ISO/IEC 27001, ki je posvečeno upravljanju incidentov iz področja informacijske varnosti, njegov

ekvivalent iz ITIL pa je proces Upravljanje incidentov, ki pa pokriva upravljanje incidentov iz vseh področij, ne samo iz informacijske varnosti, torej spet pokriva bistveno več. K A.16 lahko pripišemo tudi procesa Izpolnitev zahtev in Upravljanje problemov.

7.3.3 Predlog (poenostavljenega) modela implementacije

V začetku tega magistrskega dela sem si kot cilj zadal poiskati način, kako bi lahko podjetje, ki še nima certifikata ISO/IEC 27001, je pa PCI DSS in ITIL skladno, le-tega pridobilo na način, da porabi vse prednosti integracije standardov. V tem razdelku to skušam prikazati, pri čemer želim še posebej poudariti doseganje določene zrelosti združbe pri informacijski varnosti pred uporabo tega poenostavljenega modela.

Za predlagani model sem pripravil preglednice z natančnimi preslikavami standardov/ogrodij, ki sem jih podrobno predstavil v prejšnjih razdelkih tega poglavja. Bistvene so preglednice z oznakami (*Tabela 8 - Identificirani procesi ITIL z vsebino varovanja informacij s pripadajočimi področji*, *Tabela 12 - Visokonivojska preslikava zahtev/procesov med standardom ISO/IEC 2701 in ITIL* Vir: [72] ter *Preslikava zahtev med ISO/IEC 27001 Annex A in PCI DSS 3.1* v dodatku), na katerih sloni predlagani model.

V nadaljevanju predlagam enostavni model implementacije ISO/IEC 27001 na podlagi PCI DSS in ITIL v naslednjih korakih:

- Implementiraj zahteve iz standarda PCI DSS; v tem koraku je potrebno udejanjiti vse zahteve iz standarda PCI DSS, kot jih le-ta zahteva
- Implementiraj zahteve iz ITIL deloma ali v celoti, s certifikatom ali le za izboljšanje upravljanja storitev, odvisno od potrebe podjetja z izjemo poglavij, ki se dotikajo (tudi) informacijske varnosti (podane v tabeli 8), te morajo biti udejanjene v celoti
- Za poglavja (zahteve) ITIL, ki naj se jih udejanji, uporabi preglednico identificiranih procesov ITIL z vsebino varovanja informacij (Tabela 8)
- Začni z implementacijo ISO/IEC 27001, ko podjetje uspešno pridobi certifikat PCI DSS in ko že doseže določeno zrelost pri informacijski varnosti
- S pomočjo preglednice iz dodatka 10.2 poišči vse zahteve ISO/IEC 27001, ki imajo preslikane enako pomenske zahteve iz PCI DSS; preveri, če jih v celoti pokrivajo
- S pomočjo preglednice iz tabele 12 poišči vse zahteve ISO/IEC 27001, ki imajo preslikane enako pomenske zahteve iz ITIL; preveri, če jih v celoti pokrivajo
- Implementiraj vse zahteve ISO/IEC 27001, ki niso ali niso v celoti pokrite že s standardom PCI DSS in/ali ITIL

Na ta način bi lahko sistematično, na zahtevo natančno preverili stanje že implementiranega; v kar največji meri se izloči morebitno podvajanje in ponovna implementacija že udejanjenih zahtev. Zagotovili bi lahko zmanjšanje stroškov virov predvsem strojne in programske opreme, pa tudi človeških in ostalih virov. Tako bi se zmanjšal tudi nivo tveganj neuspešne implementacije.

Implementacija na podlagi preslikanih zahtev ne pomeni, da samo odkljukamo, ča najdemo zahtevo v tabeli, ampak si je kljub temu potrebno vsako zahtevo podrobno pojasniti; bistvo predlaganega modela je pomoč pri implementaciji novega standarda.

7.3.4 Primeri preslikav PCI DSS in ITIL zahtev v ISO/IEC 27001

V poglavju 3.4.6 je bilo prepoznanih nekaj tveganj s področja IT procesa razvoja spletne aplikacije in poslovnega procesa uporabe spletne aplikacije, ki sem jih opisal in narisal v ArchiMate modelirnem jeziku. Z ta tveganja so bile v tabeli 2 v poglavju 3.4.6.3 najdene zahteve oziroma poglavja iz ogrodja ITIL in standarda PCI DSS, ki ta tveganja ublažijo.

V nadaljevanju tega razdelka bom za vsa tveganja poskusil poiskati oziroma najti ustrezno ublažitev in ekvivalentno zahtevo še iz standarda ISO/IEC 27001. Za ta namen bom uporabil tabelo preslikav

zahtev PCI DSS 3.1 v ISO/IEC 27001:2013, ki sem jo pripravil v sklopu tega magistrskega dela (v dodatku 10.2). Prav tako bom poskušal poiskati zahteve, pokrite z ogrodjem ITIL s pomočjo tabele 12. Če preslikave v tabelah ni moč najti, potem morda standard ISO/IEC 27001 določenega tveganja ali ranljivosti ne pokriva; spet gre le za pomoč, v vsakem primeru je potrebno preveriti tudi v samem standardu neposredno.

Rezultat analize je prikazan v tabeli 13 in opisan v nadaljevanju.

Tveganje	Ogrodje/standard in poglavje, ki rešuje tveganje/ranljivost	ISO/IEC 27001 standard
IT proces razvoja spletne aplikacije		
T1.1. Napaka v programski kodi	ITIL / ISO/IEC 2000: Change management	A.12.1.2 Change management
	PCI DSS poglavje 6.3.2 6.3.2 Review custom code prior to release to production or customers in order to identify any potential coding vulnerability (using either manual or automated processes) to include at least the following:	A.14.1.1 Information security requirements analysis and specification
T1.2. Manko varnostnega testiranja	PCI DSS poglavje 11.3 (11.3.1, 11.3.2, 11.3.3, 11.3.4) 11.3 Implement a methodology for penetration testing that includes the following:	A.14.2.8 System security testing A.12.6.1 Management of technical vulnerabilities
T1.3. Namestitev nepravilne verzije aplikacije	ITIL / ISO/IEC 20000: Change management	A.12.1.2 Change management
Poslovni proces uporabe spletne aplikacije		
T2.1. Nepooblaščeni vpogledi v podatke o imetnikih plačilnih kartic (grožnja z notranje strani)	PCI DSS poglavje 2.2 2.2 Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.	/
	PCI DSS poglavje 7.1 in 7.2 7.1 Limit access to system components and cardholder data to only those individuals whose job requires such access. 7.2 Establish an access control system for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed.	A.9.1.1 Access control policy
	PCI DSS poglavje 8.1 (8.1.4) in 8.5 8.1 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:	A.9.2.1 User registration and de-registration A.9.1.1 Access control policy

	8.5 Do not use group, shared, or generic IDs, passwords, or other authentication methods as follows:	
	PCI DSS poglavje 10.1 10.1 Implement audit trails to link all access to system components to each individual user. PCI DSS poglavje 10.2 in 10.3 10.2 Implement automated audit trails for all system components to reconstruct the following events: 10.3 Record at least the following audit trail entries for all system components for each event: PCI DSS poglavje 10.7 10.7 Retain audit trail history for at least one year, with a minimum of three months immediately available for analysis (for example, online, archived, or restorable from backup).	A.12.4.1 Event logging
	PCI DSS poglavje 10.5 10.5 Secure audit trails so they cannot be altered.	A.12.4.2 Protection of log information
T2.2. Poskus napada z SQL vrivanjem (*SQL injection*)	PCI DSS poglavje 6.3 6.3 Develop internal and external software applications (including web-based administrative access to applications) securely, as follows: PCI DSS poglavje 6.3.2 6.3.2 Review custom code prior to release to production or customers in order to identify any potential coding vulnerability (using either manual or automated processes) to include at least the following:	A.14.1.1 Information security requirements analysis and specification
	PCI DSS poglavje 6.5 6.5 Address common coding vulnerabilities in software-development processes as follows:	/
	PCI DSS poglavje 6.5.1	/
	PCI DSS poglavje 6.5.2 – 6.5.10	/
	PCI DSS poglavje 6.6 6.6 For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods:	A.13.1.1 Network controls

Tabela 13 - Primeri preslikav zahtev PCI DSS in ITIL v ISO/IEC 27001

Ranljivost T1.1., napaka v programski kodi, lahko odpravimo z udejanjenjem upravljanja sprememb iz ITIL. Njegov ekvivalent iz standarda ISO/IEC 27001 je podpoglavje A.12.1.2, ki je prejšnjemu v svojih zahtevah zelo soroden. Znotraj širšega poglavja A.12 je še nekaj podpoglavij, ki so sorodni še nekaterim procesom v ITIL. S poglavjem 6.3.2 PCI DSS (pregled kode pred izdajo v produkcijo z uporabo avtomatskih in ročnih procesov) lahko povežemo vzporednice z razdelkom A.14.1.1, ki sicer neposredno ne zahteva izvajanja pregleda kode, pač pa splošno veli, naj bodo uporabljene zahteve in tehnike informacijski varnosti, med katere prav gotovo sodi tudi pregled kode.

Ranljivost T1.2, manko varnostnega testiranja lahko ublažimo z zahtevami iz (celotnega) poglavja 11.3 PCI DSS. Ekvivalent ISO/IEC 27001 sta razdelka A.14.2.8, ki zahteva varnostno testiranje sistema med razvojem in A.12.6.1, upravljanje tehničnih ranljivosti.

Zadnja iz prvega sklopa obravnavanih ranljivosti, ki se nanašajo na IT proces razvoja spletne aplikacije, je namestitev nepravilne verzije aplikacije (T1.3.), ki jo lahko ublažimo s procesom upravljanja sprememb, kateremu je, kot že omenjeno pri ranljivosti T1.1., ekvivalent podpoglavje A.12.1.2 iz ISO/IEC 27001.

Iz drugega sklopa identificiranih ranljivosti, ki se nanašajo na uporabo spletne aplikacije, je prva najdena ranljivost T2.1, ki opisuje nepooblaščene vpoglede v podatke o imetnikih plačilnih kartic, ki jih lahko izvajajo zaposleni. V izogib oziroma ublažitev temu standard PCI DSS nudi precej rešitev. Kot je razvidno iz zgornje tabele, za poglavje 2.2 ni ustreznega ekvivalenta iz ISO/IEC 27001. Poglavju 7.1 in 7.2, ki do potankosti zahtevata, kaj mora podjetje narediti glede dostopa do komponent sistema, njegov najden ekvivalent A.9.1.1 pa (ohlapno) podaja le, da naj bo politika dostopa glede na zahteve postavljena, dokumentirana in pregledana. Podobno lahko za poglavji 8.1 in 8.5 identificiramo poglavji A.9.2.1 in A.9.1.1 iz standarda ISO/IEC 27001. PCI DSS poglavja 10.1, 10.2, 10.3 in 10.7 podajajo natančne zahteve glede beleženja zgodovine in revizijske sledi. Za slednja poglavja je naj sorodnejši identificirani ekvivalent razdelek A.12.4.1, ki zahteva, naj bodo uporabniške aktivnosti, izjeme, napake in dogodki s področja informacijske varnosti beležene, hranjene in redno pregledane. Spet je opaziti manko konkretnosti pri teh zahtevah, za razliko od poglavja PCI DSS 10.5, ki enakovredno z poglavjem A.12.4.2 zahteva varovanje sledi in log zapisov.

Zadnja ranljivost je poskus napada z SQL vrivanjem (T2.2.). Iz PCI DSS sem za ublažitev te ranljivosti identificiral poglavji 6.3 in 6.3.2, katerima sorodno je poglavje A.14.1.1 (že omenjeno zgoraj). Prav tako lahko to ranljivost ublažijo rešitve iz poglavja 6.5 (ki nimajo ekvivalenta v ISO/IEC 27001, kar niti ni presenetljivo, saj so zelo konkretna v svojih zahtevah) in poglavje 6.6, ki pa mu lahko (deloma) najdemo ekvivalenta v poglavju A.13.1.1, ki predvideva, da naj bodo omrežja upravljanja na način, da ščitijo informacije v sistemih in aplikacijah.

Kot je lahko vidno iz zgornje analize in natančnejše primerjave zahtev med standardi, je PCI DSS precej bolj tog in konkreten, pa tudi morda bolj strog v zahtevah od ISO/IEC 27001 ali ITIL procesov. ISO/IEC 27001 pusti precej več manevrskega prostora in možnih interpretacij določenih zahtev, pa čeprav v svojem bistvu zajema praktično vsa področja informacijske varnosti.

8 Zaključek in nadaljnje delo

Informacijska varnost ima pomembno vlogo pri zaščiti podatkov in sredstev združbe. Slednje se morajo v celoti zavedati, da je za zaščito informacijskih sredstev potrebno nameniti čedalje več virov. Informacijska varnost mora postati ena od temeljnih skrbi slehernega podjetja ali združbe. Pri varovanju informacij (lahko) koristijo številni standardi, ogrodja, primeri dobrih praks in podobno, še posebej, če jih podjetje udejanji na kakovosten način. Res je, da je za pridobitev certifikata potrebno opraviti presojo skladnosti, kar do določene mere združbi zagotavlja določeno mero varnosti in kakovosti implementacije zahtev, vendar bi združbe morale stremeti k temu, da se pri tem nenehno izboljšujejo. Kakovostna implementacija in nenehno izboljševanje zahtevata mnogo različnih virov, tako človeških kot tudi računalniških, finančnih itd. V primeru, ko so združbe obvezane (ali pa to želijo) pridobiti več različnih certifikatov, lahko ti stroški postanejo (pre)veliki. Z integracijo in ponovno uporabo udejanjenih zahtev iz enega ali več standardov je možno stroške zmanjšati.

V magistrskem delu so bili opredeljeni koncepti varnosti, v modelirnem jeziku ArchiMate pa predstavljeni in prikazani primeri nekaterih varnostnih ranljivosti, ki sem jih identificiral pri razvoju in uporabi spletne aplikacije v združbi. Na podlagi teh primerov je bila predstavljena implementacija zahtev iz standardov PCI DSS in ITIL.

Standarda oziroma ogrodji, ki ju je v tem magistrskem delu predstavljeno podjetje uspešno vpeljalo, PCI DSS in ITIL (ISO/IEC 20000) imata različna poudarke in s tem različna cilja. S prvim lahko udejanjimo varnostne kontrole, s katerimi (predvsem) varujemo podatke o imetnikih plačilnih kartic. Z implementacijo ITIL pa si združba zagotovi kakovostno upravljanje storitev ter z njo (lahko) pridobi certifikat ISO/IEC 20000. PCI DSS je tog, zahteve so natančno opredeljene, namenjen je predvsem industriji procesiranja prometa plačilnih kartic in s tem ožjemu naboru ciljnih podjetij. To pa ne velja za ITIL, ki je namenjen vsem podjetjem, ki na trgu nudijo storitve in jih želijo nenehno izboljševati. ITIL se v svojih zahtevah ne posveča zgolj informacijski varnosti, ampak predvsem, kot že omenjeno, zagotavljanju storitev (iskanju strategije, načrtovanju, prenosu, delovanju in stalnemu izboljševanju le-teh).

Standard ISO/IEC 27001, ki ga podjetje iz te magistrske naloge (še) nima in gre torej za ciljni standard, je strogo tehničen z namenom varovanja informacij in kot tak uporaben v vsaki združbi, ki želi zaščititi svoje informacije. Zahteve so podane ohlapno in prepuščene interpretaciji implementatorja.

V magistrskem delu so bile prikazane ključne prednosti integracije standardov in ponovne uporabe že implementiranih zahtev iz enega standarda v drugega. Želel sem tudi na izbranih primerih, modeliranih v ArchiMate, pokazati, da je možno ponovno uporabiti že obstoječe rešitve. Identificirane so bile prednosti, kot so zmanjšanje stroškov za dosego določenega standarda na podlagi že udejanjenega drugega standarda, vsebinsko dopolnjevanje in izboljševanje med standardi ter zmanjšanje nivoja tveganj neuspešne implementacije.

Ugotovljeno je bilo, da so ITIL, PCI DSS in ISO/IEC 27001 precej združljivi in se lahko dobro integrirajo, s čimer lahko dobimo še izboljšan sistem varovanja informacij, ki nudi kakovost in varnost pri poslovnih procesih in storitvah, kar poleg uspešnega poslovanja lahko pomeni tudi večje zadovoljstvo strank.

V sklopu magistrskega dela je bila narejena do zahteve natančna primerjalna tabela med standardoma PCI DSS in ISO 27001, ki je bil osnova za integracijo zahtev med standardoma.

Predlagan je bil enostaven model oziroma navodila, kako se lotiti implementacije standarda ISO/IEC 27001 na podlagi že udejanjenih zahtev iz PCI DSS in ITIL. Pri tem predlogu so standardi, na katerih je možno uporabiti ta model, vnaprej točno določeni. Tak model je uporaben na primeru podjetja v tej magistrski nalogi, kjer je ciljni standard ISO/IEC 27001, že implementirana pa sta PCI DSS in ITIL.

Uporabnost predlaganega modela širše, v drugih združbah, je sicer vprašljiva, saj ima vsaka združba svoje potrebe po varnosti, zahtevanih in že udejanjenih standardih. Tak model tako ni širše uporaben, uporaben je samo v podani kombinaciji, lahko pa bi se ga uporabilo tudi brez ITIL.

Kot nadaljnje delo na področju integracije standardov informacijske varnosti bi zato morda lahko v tem magistrskem delu predlagani model implementacije standarda ISO/IEC 27001 na podlagi PCI DSS in ITIL razširil še na druge standarde, ogrodja in dobre prakse s področja varovanja informacij ter njihove kombinacije. Prav tako bi veljalo poiskati predlog za nek splošen tovrsten model, neodvisno od standarda. V nadaljevanju na višjem nivoju, v obliki osnutka v alinejah podajam predlog za omenjeni splošni model integracije oziroma implementacije enega standarda informacijske varnosti na podlagi drugega, kot podlago za nadaljnje delo:

- Preuči prvi standard/dobre prakse/ogrodje (v nadaljevanju prvi standard) s posebno pozornostjo na njegovih vsebinskih poudarkih.
- Preuči drugi standard/dobre prakse/ogrodje (v nadaljevanju drugi, ciljni standard) s posebno pozornostjo na njegovih vsebinskih poudarkih.
- Implementiraj prvi standard, s posebnim poudarkom na vsebini, ki ga v obliki kontrol zahteva tudi ciljni standard.
- Čim bolj generaliziraj implementacijo kontrol in dokumentacije prvega standarda, da je uporaben tudi izven predefiniranega obsega.
- S ciljem vzpostavitve kar se da dobro organiziranega in kakovostnega sistema za upravljanje varovanja informacij naj bodo vse zahteve iz prvega standarda obvezne.
- Začni z implementacijo drugega standarda, najraje ko je v združbi že dosežena določena zrelost informacijske varnosti (s kontrolami iz prvega standarda).
- Poišči enako pomenske kontrole med obema standardoma (*mapping*), pri čemer uporabi že vse implementirano iz prvega standarda.
- Ker je pomembno, da so zaposleni v združbi čim bolj osveščeni o informacijski varnosti, v obeh standardih poišči tovrstna poglavja/kontrole in jih uporabi kot dobre prakse.
- Iz obeh standardov poišči zahteve za razvoj infrastrukture informacijske tehnologije, s čimer se lahko doseže nenehne izboljšave in skladnost s standardi.

9 Viri in literatura

[1] T. Bradley, J. D. Burton Jr., A. Chuvakin, A. Elberg, B. Freedman, D. King, S. Paladino, and P. Shcooping, *PCI Compliance: Implementing Effective PCI Data Security Standards*. Syngress Publishing, Inc., 2007.

[2] E. A. Morse and V. Raval, "PCI DSS: Payment card industry data securty standards in context," *Computer Law & Security Repor*, vol. 2008, no. 24, pp. 540–554.

[3] M. Siponen and R. Willison, "Information security management standards: Problems and solutions," *Information & Management*, vol. 2009, no. 46, pp. 267–270.

[4] J. Hintzbergen, K. Hintzbergen, A. Smulders, and H. B. Smulders, *Foundations of Information Security Based on ISO 27001 and ISO 27002*, 3rd ed. Van Haren Publishing, Zaltbommel, www.vanharen.net.

[5] H. Susanto, M. N. Almunawar, and Y. C. Tuan, "Information Security Management System Standards: A Comparative Study of the Big Five," *International Journal of Electrical & Computer Sciences IJECS-IJENS*, vol. 11, no. 5, Oct. 2011.

[6] T. Mataracioglu, "Comparison of PCI DSS and ISO/IEC 27001 Standards," *ISACA JOURNAL*, vol. 2016, no. Vol. 1.

[7] Z. Lovrić, "Model of Simplified Implementation of PCI DSS by Using ISO 27001 Standard," presented at the Central European Conference on Information and Intelligent Systems, 2012, pp. 347–351.

[8] M. Blount, "Compilance Standards in Data Security: Why PCI DSS and ISO/IEC 27001 Should Be Integrated." Georgia Institute of Technology, College of Computing & Fortress Inc., 30-Apr-2010.

[9] S. Mubashir Ali, T. Rahim Soomro, and M. Nawaz Brohi, "Mapping Information Technology Infrastructure Library with Other Information Technology Standards and Best Practices," *Jounral of Computer Science*, vol. 2013, no. 9, pp. 1190–1196.

[10] "Technopedia," *Technopedia*. .

[11] M. MacCarthy, "Information Security Policy in the U.S. Retail Payments Industry," in *Workshop on the Economics of Information Security*, 2010.

[12] J. Liu, Y. Xiao, H. Chen, S. Ozdemir, S. Dodle, and V. Singh, "A Survey of Payment Card Industry Data Security Standard," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 12, no. 3, pp. 287–303, 2010.

[13] "Anatomy of Transaction." MasterCard.

[14] Bankart, d. o. o., "Bankart - splošna predstavitev." Feb-2013.

[15] NKBM, "Kaj prinaša Sepa posameznikom in podjetnikom?" .

[16] Bankart, "Interna dokumentacija podjetja." 2015.

[17] D. Porenta, "Razvoj spletnih storitev v Javi," Diplomsko delo, Univerza v Ljubljani, Fakultetaa za računalništvo in infoormatiko, 2012.

[18] R. de Oliveira Albuquerque, L. J. García Villalba, A. L. Sandoval Orozco, F. Buiati, and T.-H. Kim, "A Layered Trust Information Security Architecture," *Sensors*, vol. 2014, no. 14, pp. 22754–22772.

[19] S. Taubenberger and J. Jürjens, "IT Security Risk Analysis based on Business Process Models enhanced with Security Requirements."

[20] W. Al-Ahmad and B. Mohammad, "Addressing Information Security Risks by Adopting Standards," *INTERNATIONAL JOURNAL OF INFORMATION SECURITY SCIENCE*, no. 2, pp. 28–43.

[21] The Government of the Hong Kong Special Administrative Region, "An Overview of Information Security Standards." Feb-2008.

[22] K. Subrahmanyam, M. Haritha, V. Tejaswini, C. Balaram, and C. Dheeraj, "Information Security and Risk Management for Banking System," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 10, no. 3, 2014.

[23] S. Halliday, K. Badenhorst, and R. von Solms, "A business approach to effective information technology risk analysis and management," *Information Management & Computer Security*, vol. 4, no. 1, p. 19.31, 1996.

[24] A. Rot, “Enterprise Information Technology Security: Risk Management Perspective,” in *Proceedings of the World Congress on Engineering and Computer Science 2009*, 2009.
[25] D. Maček, I. Magdalenić, and N. Ivković, “Risk Assessment of the Bank’s Noncompliance with Payment Card Industry Data Security Standard,” presented at the Central European Conference on Information and Intelligent Systems, vol. 2012.
[26] “Obvladovanje informatike v poslovnih sistemih.” .
[27] A. Asosheh, P. Hajinazari, and H. Khodkari, “A Practical Implementation of ISMS,” *International Journal of Information Science and Management*, vol. 2013, no. Special Issue (ECDC 2013).
[28] G. Disterer, “ISO/IEC 27000, 27001 and 27002 for Information Security Management,” *Journal of Information Security*, vol. 2013, no. 4, pp. 92–100, Apr. 2013.
[29] G. Mirela and D. M. Boldeau, “INFORMATION SECURITY MANAGEMENT SYSTEM.”
[30] J. S. Broderick, “ISMS, security standards and security regulations,” *Information Security Technical Report*, vol. II, pp. 26–31, 2006.
[31] “Standardi sistemov za upravljanje varovanja informacij.” Housing.
[32] H. Jonkers, “Enterprise Architecture-Based Risk Assessment with ArchiMate,” *BiZZdesign*. .
[33] “ArchiMate® 2.1 Specification, Open Group Standard (C13L).” The Open Group, 2013.
[34] The Open Group, “Modeling Enterprise Risk Management and Security with the ArchiMate® Language.” The Open Group, 2015.
[35] M. Lankhorst, *Enterprise Modelling, Communication and Analysis*, 2nd ed., vol. 2009. Springer.
[36] “Archimate skozi primer.”
[37] H. Cholez and C. Feltus, “Towards an Innovative Systemic Approach of Risk Management.”
[38] F. Alisherov A., and F. Sattarova Y., “Methodology for Penetration Testing,” *International Journal of of Grid and Distributed Computing*, vol. 2, no. 2, 2009.
[39] “Payment Card Industry (PCI) Data Security Standard - Requirements and Security Assessment Procedures, Version 3.1.” PCI Security Standards Council, LLC, Apr-2015.
[40] G. Ataya, “PCI DSS audit and compliance,” *Information Security Technical Report*, vol. 15, pp. 138–144, 2010.
[41] M. Nicho, “Effectiveness of the PCI DSS 2.0 on Preventing Security Breaches: A Holistic perspective,” Jan. 2011.
[42] M. Chapple, “PCI DSS for Database Professionals,” *http://databases.about.com*. .
[43] J. Woo, L. Sael, and C. Zoltowski, “Database Auditing.”
[44] J. Wheatman and M. Nicolett, “Database Activity Monitoring Market Overview,” *Gartner*, 2009.
[45] R. Mogull, “Understanding and Selecting a Database Activity Monitoring Solution.” The SANS Institute.
[46] J. Fonseca, M. Vieira, and H. Madeira, “Online Detection of Malicious Data Access Using DBMS Auditing,” pp. 1013–1020, 2008.
[47] “MSDN: CONTEXT_INFO (Transact-SQL).” .
[48] “SQL Injection Prevention Cheat Sheet.” OWASP.org, 05-Nov-2015.
[49] “MSDN: SqlCommand.Parameters Property.” .
[50] “Historical record of ISO membership since its creation (1947).” .
[51] D. Topalovic, “ITIL and ISO 20000: A Comparison,” *ITIL & ISO 20000 Blog*. .
[52] itSM CENTER, *ITIL v3 Foundation*. itSM CENTER.
[53] M. Vicente, N. Gama, and M. Mia da Silva, “Using ArchiMate to Represent ITIL Matamodel,” 2013.
[54] “http://searchcio.techtarget.com/definition/ITSM,” *Techtarget*. .
[55] D. Favelle, “ITIL V3 - Where’s the value?” .
[56] “ITIL 2011 edition Processes along the Service Lifecycle Diagram,” *ITIL Blues*. .
[57] Taruu LLC, “ITIL® v3 Foundation Study Guide, Release Version 4.2.2.5.” Taruu.
[58] “The ISO/IEC 27000 Family of Information Security Standards,” *IT Governance*. .
[59] “ISO/IEC 27001 - Information technology - Security techniques - Information security management systems - Requirements.” ISO/IEC, 2013.
[60] D. Tunçalp, “Diffusion and Adoption of Information Security Management Standards Across Countries and Industries,” *Journal of Global Information Technology Management*, vol. 2014, no. 17:, pp. 221–227.

[61] R. Sheikhpou and N. Modiri, “A best practice approach for integration of ITIL and ISO/IEC 27001 services for information security management,” *Indian Journal of Science and Technology*, vol. 5, no. 2, pp. 2170–2176.
[62] K. V. Warren, “Security Controls in Service Management,” Dec. 2010.
[63] S. Ramanauskaite, D. Olifer, N. Goranin, and L. Radvilavičius, “Visualization of Mapped Security Standards for Analysis and Use Optimisation,” *International Journal of Computer Theory and Engineering*, vol. 6, no. 5, pp. 372–376, Oct. 2014.
[64] M. Gehrmann, “ISO/IEC 27002 for structuring comprehensive information technology for management in organizations.”
[65] S. Mubashir Ali and T. Rahim Soomro, “Integration of Information Security Essential Controls into Information Technology Infrastructure Library – A Proposed Framework,” *International Journal of Applied Science and Technology*, vol. 4, no. 1, Jan. 2014.
[66] M. Sykes and N. Landman, “ITIL and ISO/IEC 27001 - How ITIL can be used to support the delivery of compliant practices for Information Security Management Systems.” Fox IT SM Resourcing Ltd in QT&C Group Ltd, 2013.
[67] S. Wright, “Using ISO 27001 for PCI DSS Compliance.” Insight Consulting.
[68] “Mapping between the requirements of ISO/IEC 27001:2005 and ISO/IEC 27001:2013.” bsigroup.com.
[69] “Best Practices for Maintaining PCI DSS Compliance.” .
[70] M. Hofherr, “Mapping ISO27001<>PCI DSS 2.0.” .
[71] “Payment Card Industry (PCI) Data Security Standard - Summary of Changes from PCI DSS Version 2.0 to 3.0.” PCI Security Standards Council, LLC, Nov-2013.
[72] “Payment Card Industry (PCI) Data Security Standard - Summary of Changes from PCI DSS Version 3.0 to 3.1.” PCI Security Standards Council, LLC, Apr-2015.
[73] A. Segovia, “How to implement ISO 27001 and ISO 20000 together,” 16-Mar-2015. .

10 Dodatek

10.1 Preslikava zahtev med ISO/IEC 27001 Annex A in PCI DSS 3.1

ISO/IEC 27001 Annex A (2013)	ISO/IEC 27001 Annex A (2005)	PCI DSS 2.0	PCI DSS 3.1
A.5 Information security policies			
A.5.1 Management direction for information security			
A.5.1.1 Policies for information security	5.1.1	12.1	12.1 Establish, publish, maintain, and disseminate a security policy.
A.5.1.2 Review of the policies for information security	5.1.2	12.1.3	12.1.1 Review the security policy at least annually and update the policy when the environment changes.
A.6: Organization of information security			
A.6.1 Internal organization			
A.6.1.1 Information security roles and responsibilities	6.1.3	12.5	12.5 Assign to an individual or team the following information security management responsibilities:
		12.5.1	12.5.1 Establish, document, and distribute security policies and procedures.
		12.5.2	12.5.2 Monitor and analyze security alerts and information, and distribute to appropriate personnel.
		12.5.3	12.5.3 Establish, document, and distribute security incident response and escalation procedures to ensure timely and effective handling of all situations.
		12.5.4	12.5.4 Administer user accounts, including additions, deletions, and modifications.
		12.5.5	12.5.5 Monitor and control all access to data.
	8.1.1	12.4	12.4 Ensure that the security policy and procedures clearly define information security responsibilities for all personnel.
A.6.1.2 Segregation of duties	10.1.3	6.4.2	6.4.2 Separation of duties between development/test and production environments
A.6.1.3 Contact with authorities	6.1.6	/	/
A.6.1.4 Contact with special interest groups	6.1.7	/	/
A.6.1.5 Information security in project management	NEW		[Many]
A.6.2 Mobile devices and teleworking			
A.6.2.1 Mobile device policy	11.7.1	/	/
A.6.2.2 Teleworking	11.7.2	/	/

A.7: Human resource security - 6 controls that are applied before, during, or after employment			
A.7.1 Prior to employment			
A.7.1.1 Screening	8.1.2	12.7	12.7 Screen potential personnel prior to hire to minimize the risk of attacks from internal sources. (Examples of background checks include previous employment history, criminal record, credit history, and reference checks.)
A.7.1.2 Terms and conditions of employment	8.1.3	/	/
A.7.2 During employment			
A.7.2.1 Management responsibilities	8.2.1	/	
A.7.2.2 Information security awareness, education and training	8.2.2	12.6	12.6 Implement a formal security awareness program to make all personnel aware of the importance of cardholder data security.
		12.6.1	12.6.1 Educate personnel upon hire and at least annually.
		12.6.2	12.6.2 Require personnel to acknowledge at least annually that they have read and understood the security policy and procedures.
		12.9.4	12.10.4 Provide appropriate training to staff with security breach response responsibilities.
A.7.2.3 Disciplinary process	8.2.3	/	/
A.7.3 Termination and change of employment			
A.7.3.1 Termination or change of employment responsibilities	8.3.1	/	/
A.8: Asset management			
A.8.1 Responsibility for assets			
A.8.1.1 Inventory of assets	7.1.1	12.3.3	12.3.3 A list of all such devices and personnel with access
A.8.1.2 Ownership of assets	7.1.2	12.3.4	12.3.4 A method to accurately and readily determine owner, contact information, and purpose (for example, labeling, coding, and/or inventorying of devices)
A.8.1.3 Acceptable use of assets	7.1.3	12.3	12.3 Develop usage policies for critical technologies and define proper use of these technologies.
		12.3.1	12.3.1 Explicit approval by authorized parties
		12.3.5	12.3.5 Acceptable uses of the technology
		12.3.6	12.3.6 Acceptable network locations for the technologies
		12.3.7	12.3.7 List of company-approved products
A.8.1.4 Return of assets	8.3.2	/	/
A.8.2 Information classification			

A.8.2.1 Classification of information	7.2.1	9.7.1	9.6.1 Classify media so the sensitivity of the data can be determined.
A.8.2.2 Labelling of information	7.2.2	/	/
A.8.2.3 Handling of assets	10.7.3	9.6	9.5 Physically secure all media.
		9.7	9.6 Maintain strict control over the internal or external distribution of any kind of media, including the following:
		9.8	9.6.3 Ensure management approves any and all media that is moved from a secured area (including when media is distributed to individuals).
		9.9	9.7 Maintain strict control over the storage and accessibility of media.
		9.9.1	9.7.1 Properly maintain inventory logs of all media and conduct media inventories at least annually.
		12.3.10	12.3.10 For personnel accessing cardholder data via remote-access technologies, prohibit the copying, moving, and storage of cardholder data onto local hard drives and removable electronic media, unless explicitly authorized for a defined business need. Where there is an authorized business need, the usage policies must require the data be protected in accordance with all applicable PCI DSS Requirements.
		3.1	3.1 Keep cardholder data storage to a minimum by implementing data retention and disposal policies, procedures and processes that include at least the following for all cardholder data (CHD) storage:
		3.1.1	/
		3.2	3.2 Do not store sensitive authentication data after authorization (even if encrypted). If sensitive authentication data is received, render all data unrecoverable upon completion of the authorization process.
		3.2.1	3.2.1 Do not store the full contents of any track (from the magnetic stripe located on the back of a card, equivalent data contained on a chip, or elsewhere) after authorization. This data is alternatively called full track, track, track 1, track 2, and magnetic-stripe data.
		3.2.2	3.2.2 Do not store the card verification code or value (three-digit or four-digit number printed on the front or back of a payment card used to verify card-not-present transactions) after authorization.
		3.2.3	3.2.3 Do not store the personal identification number (PIN) or the encrypted PIN block after authorization.
		3.3	3.3 Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed), such that only personnel with a legitimate business need can see the full PAN.

		3.4	3.4 Render PAN unreadable anywhere it is stored (including on portable digital media, backup media, and in logs) by using any of the following approaches:
		3.4.1	3.4.1 If disk encryption is used (rather than file- or column-level database encryption), logical access must be managed separately and independently of native operating system authentication and access control mechanisms (for example, by not using local user account databases or general network login credentials). Decryption keys must not be associated with user accounts.
		9.10.2	9.8.2 Render cardholder data on electronic media unrecoverable so that cardholder data cannot be reconstructed.
A.8.3 Media handling			
A.8.3.1 Management of removable media	10.7.1	/	/
A.8.3.2 Disposal of media	10.7.2	9.10	9.8 Destroy media when it is no longer needed for business or legal reasons as follows:
		9.10.1	9.8.1 Shred, incinerate, or pulp hard-copy materials so that cardholder data cannot be reconstructed. Secure storage containers used for materials that are to be destroyed.
A.8.3.3 Physical media transfer	10.8.3	9.7.2	9.6.2 Send the media by secured courier or other delivery method that can be accurately tracked.
A.9: Access control			
A.9.1 Business requirements of access control			
A.9.1.1 Access control policy	11.1.1	2.1	2.1 Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network. This applies to ALL default passwords, including but not limited to those used by operating systems, software that provides security services, application and system accounts, point-of-sale (POS) terminals, Simple Network Management Protocol (SNMP) community strings, etc.).
		7.1	7.1 Limit access to system components and cardholder data to only those individuals whose job requires such access.
		7.1.4	/
		7.2	7.2 Establish an access control system for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed.
		7.2.1	7.2.1 Coverage of all system components
		7.2.2	7.2.2 Assignment of privileges to individuals based on job classification and function.
		7.2.3	7.2.3 Default "deny-all" setting.

		8.2	8.2 In addition to assigning a unique ID, ensure proper user-authentication management for non-consumer users and administrators on all system components by employing at least one of the following methods to authenticate all users:
		8.5	8.5 Do not use group, shared, or generic IDs, passwords, or other authentication methods as follows:
		8.5.1	8.5.1 Additional requirement for service providers only: Service providers with remote access to customer premises (for example, for support of POS systems or servers) must use a unique authentication credential (such as a password/phrase) for each customer.
		8.5.10	8.2.3 Passwords/phrases must meet the following:
		8.5.11	/
A.9.1.2 Access to networks and network services	11.4.1	1.1.4	1.1.4 Requirements for a firewall at each Internet connection and between any demilitarized zone (DMZ) and the internal network zone
A.9.2 User access management			
A.9.2.1 User registration and de-registration	11.2.1	7.1.3	7.1.3 Assign access based on individual personnel's job classification and function.
	11.5.2	8.1	8.1 Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:
		8.5.8	8.5 Do not use group, shared, or generic IDs, passwords, or other authentication methods as follows:
		12.3.2	12.3.2 Authentication for use of the technology
A.9.2.2 User access provisioning	11.2.1	7.1.3	7.1.3 Assign access based on individual personnel's job classification and function.
A.9.2.3 Management of privileged access rights	11.2.2	7.1.1	7.1.2 Restrict access to privileged user IDs to least privileges necessary to perform job responsibilities.
		7.1.2	/
A.9.2.4 Management of secret authentication information of users	11.2.3	8.5.2	8.2.2 Verify user identity before modifying any authentication credential—for example, performing password resets, provisioning new tokens, or generating new keys.
		8.5.3	/
		8.5.7	8.4 Document and communicate authentication policies and procedures to all users including:
		8.5.9	/
A.9.2.5 Review of user access rights	11.2.4	8.5.5	/
A.9.2.6 Removal or adjustment of access rights	8.3.3	8.5.4	/
A.9.3 User responsibilities			
A.9.3.1 Use of secret authentication information	11.3.1	/	/

A.9.4 System and application access control			
A.9.4.1 Information access restriction	11.6.1	8.5.16	8.7 All access to any database containing cardholder data (including access by applications, administrators, and all other users) is restricted as follows:
A.9.4.2 Secure log-on procedures	11.5.1	8.5.13	/
		8.5.14	/
	11.5.5	8.5.15	/
		12.3.8	12.3.8 Automatic disconnect of sessions for remote-access technologies after a specific period of inactivity
	11.5.6	/	/
A.9.4.3 Password management system	11.5.3	8.5.12	/
A.9.4.4 Use of privileged utility programs	11.5.4	/	/
A.9.4.5 Access control to program source code	12.4.3	/	/
A.10: Cryptography			
A.10.1 Cryptographic controls			
A.10.1.1 Policy on the use of cryptographic controls	12.3.1	8.4	8.4 Document and communicate authentication policies and procedures to all users including:
		2.3	2.3 Encrypt all non-console administrative access using strong cryptography. Use technologies such as SSH, VPN, or TLS for web-based management and other non-console administrative access.
		4.1	4.1 Use strong cryptography and security protocols (for example, TLS, IPSEC, SSH, etc.) to safeguard sensitive cardholder data during transmission over open, public networks, including the following:
A.10.1.2 Key management	12.3.2	3.5	3.5 Document and implement procedures to protect keys used to secure stored cardholder data against disclosure and misuse:
		3.5.1	3.5.1 Restrict access to cryptographic keys to the fewest number of custodians necessary.
		3.5.2	3.5.2 Store secret and private keys used to encrypt/decrypt cardholder data in one (or more) of the following forms at all times:
		3.6	3.6 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data, including the following:
		3.6.1	3.6.1 Generation of strong cryptographic keys
		3.6.2	3.6.2 Secure cryptographic key distribution
		3.6.3	3.6.3 Secure cryptographic key storage
		3.6.4	3.6.4 Cryptographic key changes for keys that have reached the end of their cryptoperiod (for example, after a defined period of time has

			passed and/or after a certain amount of cipher-text has been produced by a given key), as defined by the associated application vendor or key owner, and based on industry best practices and guidelines (for example, NIST Special Publication 800-57).
		3.6.5	3.6.5 Retirement or replacement (for example, archiving, destruction, and/or revocation) of keys as deemed necessary when the integrity of the key has been weakened (for example, departure of an employee with knowledge of a clear-text key component), or keys are suspected of being compromised.
		3.6.6	3.6.6 If manual clear-text cryptographic key-management operations are used, these operations must be managed using split knowledge and dual control.
		3.6.7	3.6.7 Prevention of unauthorized substitution of cryptographic keys.
		3.6.8	3.6.8 Requirement for cryptographic key custodians to formally acknowledge that they understand and accept their key-custodian responsibilities.
A.11 Physical and environmental security			
A.11.1 Secure areas			
A.11.1.1 Physical security perimeter	9.1.1	/	/
A.11.1.2 Physical entry controls	9.1.2	9.1	9.1 Use appropriate facility entry controls to limit and monitor physical access to systems in the cardholder data environment.
		9.1.1	9.1.1 Use video cameras and/or access control mechanisms to monitor individual physical access to sensitive areas. Review collected data and correlate with other entries. Store for at least three months, unless otherwise restricted by law.
		9.2	9.2 Develop procedures to easily distinguish between onsite personnel and visitors, to include:
		9.3	9.4 Implement procedures to identify and authorize visitors.
		9.3.1	9.4.1 Visitors are authorized before entering, and escorted at all times within, areas where cardholder data is processed or maintained.
		9.3.2	9.4.2 Visitors are identified and given a badge or other identification that expires and that visibly distinguishes the visitors from onsite personnel.
		9.3.3	9.4.3 Visitors are asked to surrender the badge or identification before leaving the facility or at the date of expiration.
		9.4	9.4.4 A visitor log is used to maintain a physical audit trail of visitor activity to the facility as well as computer rooms and data centers where cardholder data is stored or transmitted.

A.11.1.3 Securing offices, rooms and facilities	9.1.3	/	/
A.11.1.4 Protecting against external and environmental threats	9.1.4	/	/
A.11.1.5 Working in secure areas	9.1.5	/	/
A.11.1.6 Delivery and loading areas	9.1.6	/	/
A.11.2 Equipment			
A.11.2.1 Equipment siting and protection	9.2.1	9.1.3	9.1.3 Restrict physical access to wireless access points, gateways, handheld devices, networking/communications hardware, and telecommunication lines.
A.11.2.2 Supporting utilities	9.2.2	/	/
A.11.2.3 Cabling security	9.2.3	9.1.2	9.1.2 Implement physical and/or logical controls to restrict access to publicly accessible network jacks.
A.11.2.4 Equipment maintenance	9.2.4	/	/
A.11.2.5 Removal of assets	9.2.7	/	/
A.11.2.6 Security of equipment and assets off-premises	9.2.5	/	/
A.11.2.7 Secure disposal or reuse of equipment	9.2.6	/	/
A.11.2.8 Unattended user equipment	11.3.2	/	/
A.11.2.9 Clear desk and clear screen policy	11.3.3	/	/
A.12 Operations security			
A.12.1 Operational procedures and responsibilities			
A.12.1.1 Documented operating procedures	10.1.1	12.2	/
A.12.1.2 Change management	10.1.2	1.1.1	1.1.1 A formal process for approving and testing all network connections and changes to the firewall and router configurations
		6.4	6.4 Follow change control processes and procedures for all changes to system components. The processes must include the following:
		6.4.5	6.4.5 Change control procedures for the implementation of security patches and software modifications must include the following:
		6.4.5.1	6.4.5.1 Documentation of impact.
		6.4.5.2	6.4.5.2 Documented change approval by authorized parties.
		6.4.5.3	6.4.5.3 Functionality testing to verify that the change does not adversely impact the security of the system.
		6.4.5.4	6.4.5.4 Back-out procedures.

A.12.1.3 Capacity management	10.3.1	/	/
A.12.1.4 Separation of development, testing and operational environments	10.1.4	6.4.1	6.4.1 Separate development/test environments from production environments, and enforce the separation with access controls.
A.12.2 Protection from malware			
A.12.2.1 Controls against malware	10.4.1	5.1	5.1 Deploy anti-virus software on all systems commonly affected by malicious software (particularly personal computers and servers).
		5.1.1	5.1.1 Ensure that anti-virus programs are capable of detecting, removing, and protecting against all known types of malicious software.
		5.2	5.2 Ensure that all anti-virus mechanisms are maintained as follows:
	10.4.2	/	/
A.12.3 Backup			
A.12.3.1 Information backup	10.5.1	9.5	9.5.1 Store media backups in a secure location, preferably an off-site facility, such as an alternate or backup site, or a commercial storage facility. Review the location's security at least annually.
A.12.4 Logging and monitoring			
A.12.4.1 Event logging	10.10.1	A.1.3	A.1.3 Ensure logging and audit trails are enabled and unique to each entity's cardholder data environment and consistent with PCI DSS Requirement 10.
		10.1	10.1 Implement audit trails to link all access to system components to each individual user.
		10.2	10.2 Implement automated audit trails for all system components to reconstruct the following events:
		10.2.1	10.2.1 All individual user accesses to cardholder data
		10.2.2	10.2.2 All actions taken by any individual with root or administrative privileges
		10.2.3	10.2.3 Access to all audit trails
		10.2.4	10.2.4 Invalid logical access attempts
		10.2.5	10.2.5 Use of and changes to identification and authentication mechanisms—including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges
		10.2.6	10.2.6 Initialization, stopping, or pausing of the audit logs
		10.2.7	10.2.7 Creation and deletion of system-level objects
		10.3	10.3 Record at least the following audit trail entries for all system components for each event:
		10.3.1	10.3.1 User identification
		10.3.2	10.3.2 Type of event
		10.3.3	10.3.3 Date and time

		10.3.4	10.3.4 Success or failure indication
		10.3.5	10.3.5 Origination of event
		10.3.6	10.3.6 Identity or name of affected data, system component, or resource.
		10.7	10.7 Retain audit trail history for at least one year, with a minimum of three months immediately available for analysis (for example, online, archived, or restorable from backup).
	10.10.2	10.6	10.6 Review logs and security events for all system components to identify anomalies or suspicious activity.
	10.10.5		
A.12.4.2 Protection of log information	10.10.3	10.5	10.5 Secure audit trails so they cannot be altered.
		10.5.1	10.5.1 Limit viewing of audit trails to those with a job-related need.
		10.5.2	10.5.2 Protect audit trail files from unauthorized modifications.
		10.5.3	10.5.3 Promptly back up audit trail files to a centralized log server or media that is difficult to alter.
		10.5.4	10.5.4 Write logs for external-facing technologies onto a secure, centralized, internal log server or media device.
		10.5.5	10.5.5 Use file-integrity monitoring or change-detection software on logs to ensure that existing log data cannot be changed without generating alerts (although new data being added should not cause an alert).
A.12.4.3 Administrator and operator logs	10.10.3	10.5	/
		10.5.1	/
		10.5.2	/
		10.5.3	/
		10.5.4	/
		10.5.5	/
	10.10.4	/	/
A.12.4.4 Clock synchronisation	10.10.6	10.4	10.4 Using time-synchronization technology, synchronize all critical system clocks and times and ensure that the following is implemented for acquiring, distributing, and storing time.
		10.4.1	10.4.1 Critical systems have the correct and consistent time.
		10.4.2	10.4.2 Time data is protected.
		10.4.3	10.4.3 Time settings are received from industry-accepted time sources.
A.12.5 Control of operational software			
A.12.5.1 Installation of software on operational systems	12.4.1	2.2.3	2.2.3 Implement additional security features for any required services, protocols, or daemons that are considered to be insecure—for example, use secured technologies such as SSH, S-FTP, TLS,

			or IPSec VPN to protect insecure services such as NetBIOS, file-sharing, Telnet, FTP, etc.
		2.2.4	2.2.4 Configure system security parameters to prevent misuse.
		6.3.1	6.3.1 Remove development, test and/or custom application accounts, user IDs, and passwords before applications become active or are released to customers.
		6.4.4	6.4.4 Removal of test data and accounts before production systems become active
A.12.6 Technical vulnerability management			
A.12.6.1 Management of technical vulnerabilities	12.6.1	6.1	6.2 Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release.
		6.2	6.1 Establish a process to identify security vulnerabilities, using reputable outside sources for security vulnerability information, and assign a risk ranking (for example, as “high,” “medium,” or “low”) to newly discovered security vulnerabilities.
		11.1	11.1 Implement processes to test for the presence of wireless access points (802.11), and detect and identify all authorized and unauthorized wireless access points on a quarterly basis.
		11.2	11.2 Run internal and external network vulnerability scans at least quarterly and after any significant change in the network (such as new system component installations, changes in network topology, firewall rule modifications, product upgrades).
		11.2.1	11.2.1 Perform quarterly internal vulnerability scans and rescans as needed, until all “high-risk” vulnerabilities (as identified in Requirement 6.1) are resolved. Scans must be performed by qualified personnel.
		11.2.2	11.2.2 Perform quarterly external vulnerability scans, via an Approved Scanning Vendor (ASV) approved by the Payment Card Industry Security Standards Council (PCI SSC). Perform rescans as needed, until passing scans are achieved. Note: Quarterly e
		11.2.3	11.2.3 Perform internal and external scans, and rescans as needed, after any significant change. Scans must be performed by qualified personnel.
		11.3	11.3.1 Perform external penetration testing at least annually and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub-network added to the environment, or a web server added to the environment).

			11.3.2 Perform internal penetration testing at least annually and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub-network added to the environment, or a web server added to the environment).
			11.3.3 Exploitable vulnerabilities found during penetration testing are corrected and testing is repeated to verify the corrections.
A.12.6.2 Restrictions on software installation	NEW		12.3.3 A list of all such devices and personnel with access
			12.3.7 List of company-approved products
A.12.7 Information systems audit considerations			
A.12.7.1 Information systems audit controls	15.3.1	/	/
A.13: Communications security			
A.13.1 Network security management			
A.13.1.1 Network controls	10.6.1	1.1	1.1 Establish and implement firewall and router configuration standards that include the following:
		1.1.2	1.1.2 Current network diagram that identifies all connections between the cardholder data environment and other networks, including any wireless networks
			1.1.3 Current diagram that shows all cardholder data flows across systems and networks
		1.1.5	1.1.6 Documentation and business justification for use of all services, protocols, and ports allowed, including documentation of security features implemented for those protocols considered to be insecure.
		1.2.2	1.2.2 Secure and synchronize router configuration files.
		2.1.1	2.1.1 For wireless environments connected to the cardholder data environment or transmitting cardholder data, change ALL wireless vendor defaults at installation, including but not limited to default wireless encryption keys, passwords, and SNMP community strings.
		4.1.1	4.1.1 Ensure wireless networks transmitting cardholder data or connected to the cardholder data environment, use industry best practices (for example, IEEE 802.11i) to implement strong encryption for authentication and transmission.
		11.4	11.4 Use intrusion-detection and/or intrusion-prevention techniques to detect and/or prevent intrusions into the network. Monitor all traffic at the perimeter of the cardholder data environment as well as at critical points in the cardholder data

			environment, and alert personnel to suspected compromises.
		1.4	1.4 Install personal firewall software on any mobile and/or employee-owned devices that connect to the Internet when outside the network (for example, laptops used by employees), and which are also used to access the network. Firewall configurations include:
		2.2.2	2.2.2 Enable only necessary services, protocols, daemons, etc., as required for the function of the system.
			2.2.3 Implement additional security features for any required services, protocols, or daemons that are considered to be insecure—for example, use secured technologies such as SSH, S-FTP, TLS, or IPSec VPN to protect insecure services such as NetBIOS, file-sharing, Telnet, FTP, etc.
		6.6	6.6 For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods:
A.13.1.2 Security of network services	10.6.2	/	/
A.13.1.3 Segregation in networks	11.4.5	1.1.3	1.1.3 Current diagram that shows all cardholder data flows across systems and networks
		1.2	1.2 Build firewall and router configurations that restrict connections between untrusted networks and any system components in the cardholder data environment.
		1.2.1	1.2.1 Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment, and specifically deny all other traffic.
		1.2.3	1.2.3 Install perimeter firewalls between all wireless networks and the cardholder data environment, and configure these firewalls to deny or, if traffic is necessary for business purposes, permit only authorized traffic between the wireless environment and the cardholder data environment.
		1.3	1.3 Prohibit direct public access between the Internet and any system component in the cardholder data environment.
		1.3.1	1.3.1 Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.
		1.3.2	1.3.2 Limit inbound Internet traffic to IP addresses within the DMZ.
		1.3.3	1.3.3 Do not allow any direct connections inbound or outbound for traffic between the Internet and the cardholder data environment.

		1.3.4	1.3.4 Implement anti-spoofing measures to detect and block forged source IP addresses from entering the network.
		1.3.5	1.3.5 Do not allow unauthorized outbound traffic from the cardholder data environment to the Internet.
		1.3.6	1.3.6 Implement stateful inspection, also known as dynamic packet filtering. (That is, only "established" connections are allowed into the network.)
		1.3.7	1.3.7 Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.
		1.3.8	1.3.8 Do not disclose private IP addresses and routing information to unauthorized parties.
A.13.2 Information transfer			
A.13.2.1 Information transfer policies and procedures	10.8.1	/	/
A.13.2.2 Agreements on information transfer	10.8.2	/	/
A.13.2.3 Electronic messaging	10.8.4	4.2	4.2 Never send unprotected PANs by end-user messaging technologies (for example, e-mail, instant messaging, SMS, chat, etc.).
A.13.2.4 Confidentiality or nondisclosure agreements	6.1.5	/	
A.14 System acquisition, development and maintenance			
A.14.1 Security requirements of information systems			
A.14.1.1 Information security requirements analysis and specification	12.1.1	6.3	6.3 Develop internal and external software applications (including web-based administrative access to applications) securely, as follows:
		6.3.2	6.3.2 Review custom code prior to release to production or customers in order to identify any potential coding vulnerability (using either manual or automated processes) to include at least the following:
A.14.1.2 Securing application services on public networks	10.9.1	/	/
	10.9.3	/	/
A.14.1.3 Protecting application services transactions	10.9.2	/	/
A.14.2 Security in development and support processes			
A.14.2.1 Secure development policy	NEW		6.7 Ensure that security policies and operational procedures for developing and maintaining secure

			systems and applications are documented, in use, and known to all affected parties.
A.14.2.2 System change control procedures	12.5.1	/	/
A.14.2.3 Technical review of applications after operating platform changes	12.5.2	/	/
A.14.2.4 Restrictions on changes to software packages	12.5.3	/	/
A.14.2.5 Secure system engineering principles	NEW		/
A.14.2.6 Secure development environment	NEW		6.4.2 Separation of duties between development/test and production environments
			6.4.3 Production data (live PANs) are not used for testing or development
			6.4.4 Removal of test data and accounts before production systems become active
A.14.2.7 Outsourced development	12.5.5	/	/
A.14.2.8 System security testing	NEW		11.3 Implement a methodology for penetration testing that includes the following:
			11.3.1 Perform *external* penetration testing at least annually and after any significant infrastructure or application upgrade or modification
			11.3.2 Perform *internal* penetration testing at least annually and after any significant infrastructure or application upgrade
			11.3.3 Exploitable vulnerabilities found during penetration testing are corrected and testing is repeated to verify the corrections.
			11.3.4 If segmentation is used to isolate the CDE from other networks, perform penetration tests
A.14.2.9 System acceptance testing	10.3.2	/	/
A.14.3 Test data			
A.14.3.1 Protection of test data	12.4.2	6.4.3	6.4.3 Production data (live PANs) are not used for testing or development
A.15: Supplier relationships			
A.15.1 Information security in supplier relationships			
A.15.1.1 Information security policy for supplier relationships	NEW		9.9.3 Provide training for personnel to be aware of attempted tampering or replacement of devices. Training should include the following:
A.15.1.2 Addressing security within supplier agreements	6.2.3	12.8	12.8 Maintain and implement policies and procedures to manage service providers with whom cardholder data is shared, or that could affect the security of cardholder data, as follows:
		12.8.1	12.8.1 Maintain a list of service providers.

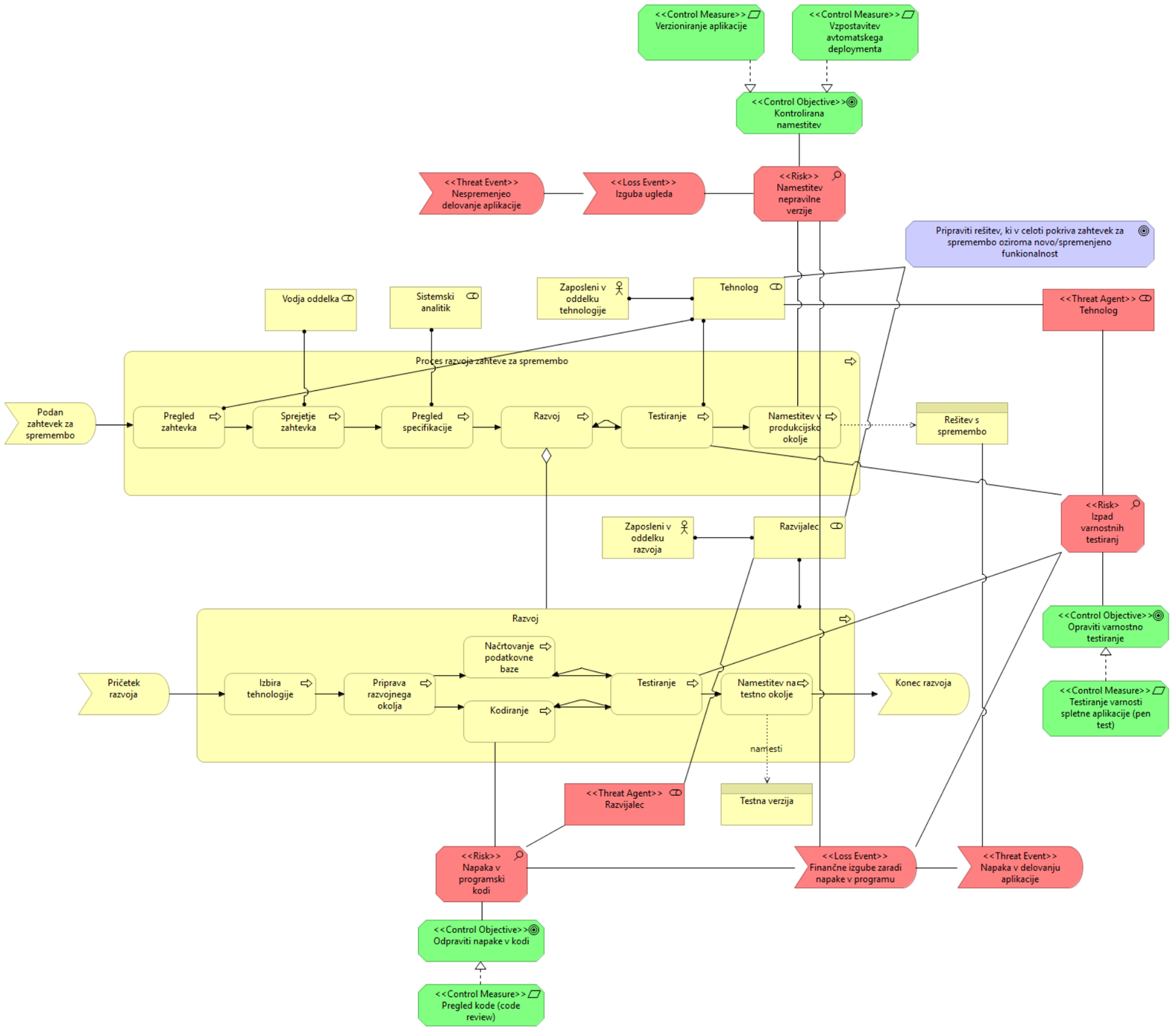
		12.8.2	12.8.2 Maintain a written agreement that includes an acknowledgement that the service providers are responsible for the security of cardholder data the service providers possess or otherwise store, process or transmit on behalf of the customer, or to the extent that they could impact the security of the customer's cardholder data environment.
		12.8.3	12.8.3 Ensure there is an established process for engaging service providers including proper due diligence prior to engagement.
A.15.1.3 Information and communication technology supply chain	NEW		9.9.3 Provide training for personnel to be aware of attempted tampering or replacement of devices. Training should include the following:
A.15.2 Supplier service delivery management			
A.15.2.1 Monitoring and review of supplier services	10.2.2	12.8.4	12.8.4 Maintain a program to monitor service providers' PCI DSS compliance status at least annually.
A.15.2.2 Managing changes to supplier services	10.2.3	/	/
A.16 Information security incident management			
A.16.1 Management of information security incidents and improvements			
A.16.1.1 Responsibilities and procedures	13.2.1	12.9	12.10 Implement an incident response plan. Be prepared to respond immediately to a system breach.
		12.9.1	12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:
		12.9.2	12.10.2 Test the plan at least annually.
		12.9.3	12.10.3 Designate specific personnel to be available on a 24/7 basis to respond to alerts.
		12.9.5	12.10.5 Include alerts from security monitoring systems, including but not limited to intrusion-detection, intrusion-prevention, firewalls, and file-integrity monitoring systems.
A.16.1.2 Reporting information security events	13.1.1	/	/
A.16.1.3 Reporting information security weaknesses	13.1.2	/	/
A.16.1.4 Assessment of and decision on information security events	NEW		10.3 Record at least the following audit trail entries for all system components for each event:
			10.6.1 Review the following at least daily:
			10.6.2 Review logs of all other system components periodically based on the organization's policies and risk management strategy, as determined by the organization's annual risk assessment.

			10.6.3 Follow up exceptions and anomalies identified during the review process.
			12.10 Implement an incident response plan. Be prepared to respond immediately to a system breach.
			12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:
A.16.1.5 Response to information security incidents	NEW		12.10 Implement an incident response plan. Be prepared to respond immediately to a system breach.
			12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:
A.16.1.6 Learning from information security	13.2.2	12.9.6	12.10.6 Develop a process to modify and evolve the incident response plan according to lessons learned and to incorporate industry developments.
A.16.1.7 Collection of evidence	13.2.3	A.1.4	A.1.4 Enable processes to provide for timely forensic investigation in the event of a compromise to any hosted merchant or service provider.
A.17 Information security aspects of business continuity management			
A.17.1 Information security continuity			
A.17.1.1 Planning information security continuity	14.1.2	/	/
A.17.1.2 Implementing information security continuity	14.1.1	/	/
	14.1.3	/	/
	14.1.4	/	/
A.17.1.3 Verify, review and evaluate information security continuity	14.1.5	/	/
A.17.2 Redundancies			
A.17.2.1 Availability of information processing facilities	NEW		9.5 Physically secure all media.
			9.5.1 Store media backups in a secure location, preferably an off-site facility, such as an alternate or backup site, or a commercial storage facility. Review the location's security at least annually.
			12.10.1 Create the incident response plan to be implemented in the event of system breach. Ensure the plan addresses the following, at a minimum:
A.18 Compliance			

A.18.1 Compliance with legal and contractual requirements			
A.18.1.1 Identification of applicable legislation and contractual requirements	15.1.1	/	/
A.18.1.2 Intellectual property rights	15.1.2	/	/
A.18.1.3 Protection of records	15.1.3	/	/
A.18.1.4 Privacy and protection of personally identifiable information	15.1.4	/	/
A.18.1.5 Regulation of cryptographic controls	15.1.6	/	/
A.18.2 Information security reviews			
A.18.2.1 Independent review of information security	6.1.8	/	/
A.18.2.2 Compliance with security policies and standards	15.2.1	12.1.1	/
A.18.2.3 Technical compliance review	15.2.2	/	/

10.2 Povečane slike modelov tveganj

10.2.1 Model tveganj pri IT procesu razvoja programske opreme



10.2.2 Model tveganj pri poslovnem procesu uporabe spletne aplikacije